Protein Fold Recognition Using Neural Networks

Thesis

Protein Fold Recognition Using Neural Networks

Kuang Lin

This thesis is submitted in partial fulfilment of the requirements of the Open University for the degree of Doctor of Philosophy

April 2003

Division of Mathematical Biology
National Institute for Medical Research
The Ridgeway, Mill Hill, London. NW7 1AA
United Kingdom

Date of submission: 9 May 2002
Date of award: 28 April 2003

Abstract

To predict accurately the three-dimensional (3D) structures of proteins from their amino acid sequences alone remains a challenging problem. However, using protein fold recognition tools, it is often possible to achieve good models or at least to gain some more information, to aid scientists in their research. This thesis describes development of TUNE (Threading Using Neural Networks), a fold recognition program using artificial neural network (ANN) models.

A new method to generate amino acid substitution matrices is described in chapter two. It uses an ANN to generalise amino acid substitutions observed in protein structure alignments. Matrices for alignment scoring from this approach were compared with classic alignment scoring schemes.

From these neural network models, a series of encoding schemes were constructed. These schemes describe the amino acid types with a few numbers. They were generated to replace the orthogonal encoding scheme, so that smaller, faster and more accurate neural network models can be applied on bioinformatic problems.

The TUNE model was introduced in chapter four to measure protein sequence-structure compatibility. Given the integrated residue structural environment descriptions, the model predicts probabilities of observing amino acid types in such environments. Using this model, a scoring function to measure the fitness of a residue in a protein structure model can be made for protein threading programs.

The model in chapter two was extended by including the residue structural environment descriptions for predictions. A simple protein fold recognition program with a dynamic programming algorithm was developed using this model. The program was then tested in the fourth round of the Critical Assessment of protein Structure Prediction methods (CASP4) and produced reasonably good results.

Acknowledgements

I would like to thank my supervisors Dr. Alex C. W. May and Dr. William R. Taylor for their help and encouragement.

I am grateful to Delmiro Fernandez-Reyes and Jose W. Saldanha for helpful discussions.

I thank my colleagues Darrel, Enrico, Franca, Inti, Jaap, Jakir, Jens, John, Loredana, Michael, Nelly, Nigel.B, Nigel.D, Richard, Robin and Victor for making the last three years enjoyable. Thanks to all the past and present members in the division.

Thanks also to the CASP4 organisers.

I dedicated this thesis to my aunt Dr. Bing Xiao for her invaluable support over the years.

Contents

List of Figures

List of Tables

Abbreviations and Programs

1D	One Dimensional
3D	Three Dimensional
AESNN	Amino Acid Encoding Schemes from Neural Networks.
ANN	Artificial Neural Network
BLAST	Basic Local Alignment Search Tool.
BLOSUM	BLOcks SUbstitution Matrix
CATH	Class, Architecture, Topology, Homologue
CASP	Critical Assessment of Protein Structure Prediction Methods.
DSSP	Dictionary of Secondary Structure of Proteins
FASTA	Global Alignment Search Tool.
GCC	GNU (GNU is Not Unix) C Compiler
GenTHREADER	Fold Recognition Method for Genomic Sequences.
HMM	Hidden Markov Model
MST	Multiple Sequence Threading
MULTAL	Multiple Alignment Tool.
NMR	Nuclear Magnetic Resonance.
PAM	Point Accepted Mutation.
PDB	Protein Data Bank
PET	Protein Exchange Table
PHD	Predict Protein at Heidelberg.
PSI-BLAST	Position-Specific Iterated BLAST
PSIPRED	Protein Secondary Structure Prediction based on PSSMs
PSSM	Position-Specific Scoring Matrix.
RNA	Ribonuleic Acid
RMSD	Root Mean Squared Deviation.
SAP	Structural Alignment of Proteins.
SCOP	Structural Classification of Proteins.
SMN	Amino Acid Substitution Matrices from Neural Networks.
TUNE	Threading Using Neural Networks.
TUNE1D	Threading Using Neural Networks with One-dimensional Profiles
SSpro	Secondary Structure Prediction Program

NB: standard one and three letter codes are used for amino acids. Brookhaven Database codes are used for PDB entries.

Original Publications

1 Lin, K., May, A. C. W. & Taylor, W. R. (2001). Amino acid substitution matrices from an artificial neural network model. *Journal of Computational Biology* **8**(5), 471-481. (chapter two)

2 Lin, K., May, A. C. W. & Taylor, W. R. (2002a). Amino acid encoding schemes from protein structural alignments: multi-dimensional vectors to describe residue types. *Journal of Theoretical Biology (in press)* (chapter three)

3 Lin, K., May, A. C. W & Taylor, W. R. (2002b). Threading using neural networks (TUNE): the measure of protein sequence-structure compatibility. *Bioinformatics (in press)* (chapter four)

Chapter 1.

Introduction

With data from large-scale genome sequencing projects, the databanks of protein sequences (e.g. SWISSPROT Bairoch and Apweiler 1996) are rapidly expanding. Bioinformatics, the application of computers in biological sciences and especially analysis of biological sequences, is becoming an essential tool in molecular biology. One of the main aims of bioinformatics is to model the molecular interactions within cells. To do this, structural knowledge of proteins (the essential active agents in biochemistry) is vital. This is because the three-dimensional (3D) structure of a protein determines its biological function (for recent reviews, see Thornton et al. 1999, Orengo et al. 2001). However, despite significant improvements in structure determination techniques (X-ray crystallography and NMR spectroscopy), solving experimental structures of proteins is still expensive and time-consuming. To aid the design of experiments, interpret molecular biological functions and improve our understanding of proteins, much software has been developed to predict different aspects of the 3D structure of a protein from its sequence of amino acid residues alone. With fast computers, these tools are being used for high-throughput structural and functional studies of proteins in structural and functional genomics.

1.1 Protein structure

1.1.1 Protein primary structure

Twenty different building blocks (amino acids) combine in a linear chain to form proteins. All of these amino acids have a central carbon atom (Cα), an amino group (NH_2) and a carboxyl group (COOH). They are joined end to end in protein synthesis by the formation of peptide bonds between amino and carboxyl groups. So, for a protein of N amino acids, there are 20^N possible sequences. Even for a small protein, exhaustive sampling of possible sequences is beyond current computing power.

It is generally assumed that a protein sequence folds to a native conformation or ensemble of conformations that is at or near the global free-energy minimum. All necessary information for a protein to fold into its native secondary and tertiary structure is coded in its amino acid sequence (Anfinsen 1973).

1.1.2 Protein 3D structure

Protein 3D structure is hierarchically organised (Honig 1999). The highest level is constituted by the complete protein, which can be subdivided through domains to secondary structures. Domains are stable, compact evolutionary units of protein structure,

which can fold autonomously and perform their own functions semi-independently (e.g. Wetlaufer, 1973, Richardson 1981, Bork 1991, Holm and Sander 1998). Protein secondary structures are continuous fragments in protein sequences showing distinct geometrical features (Ramachandran et al. 1974). The two basic secondary structures are the α helix and β strand. Their structural features, including regular patterns of hydrogen bonds between atoms, can be easily recognised (e.g. Kabsch and Sander 1983).

1.1.3 Protein structure comparison and classification

Given the protein 3D structures from experiments, Murzin et al. (1995) established SCOP (Structural Classification Of Proteins) from the PDB (Protein Structure Databank) (Abola et al. 1987, Sussman et al. 1998) and proteins with published descriptions of their structures using a unique manual approach. First, domains of protein structures are manually assigned. Domains are then hierarchically classified into classes, common folds, superfamilies and families according to structural similarities assigned via visual inspection and functional features. SCOP is still considered one of the most accurate classifications of protein structures.

However, to make large-scale classification tasks manageable, automatic tools have to be developed to compare protein structures (e.g. Taylor and Orengo 1989, Holm and Sander 1993, May and Johnson 1994). Automatic (e.g. Holm and Sander 1994) or semi-automatic (e.g. Orengo et al. 1997) classification of protein structures can be constructed with these programs.

Proteins can have considerable structural similarities even in the absence of detectable sequence similarity. It is well known that protein structure is more conserved than protein sequence (e.g. Chothia and Lesk 1986). As a more accurate indication of protein relationships, structure alignments and structure classifications are often used as the targets of protein sequence alignment and structure prediction programs (e.g. Brenner et al. 1998, Lindahl and Elofsson 2000, Cristobal et al. 2001).

1.2 Predicting protein structure

Different mathematical models have been designed to describe and simulate the evolution and folding of proteins so that we can then predict protein structures with these models.

1.2.1 Sequence similarity and scoring matrices

An interesting problem itself, measuring the extent of similarity between protein sequences is the basis of most bioinformatic analysis. So, the first step is to define a model of divergent evolution of protein sequences. Sequence alignment is the most common way to describe similarity between protein sequences. In alignment programs using the popular dynamic programming algorithm (e.g. Needleman and Wunsch 1970), the substitution of each residue is considered independently. So, the model of residue evolution directly affects the scoring of sequence alignment. Dayhoff and co-workers (1978) introduced the PAM model of amino acid substitution. In their Markov model, it was assumed that each mutational event was independent of previous events. A table of 20*20 mutation probabilities of amino acids at an evolutionary distance of 1 PAM (Point Accepted Mutation) were estimated using alignments of sequences of closely related proteins. Substitution matrices appropriate for greater evolutionary distances can then be generated by repeated multiplication of the 1 PAM matrix. From these substitution matrices, log-odds matrices were generated for the scoring of protein sequence alignment. PAM scoring matrices have been the standard scoring matrices for many

sequence alignment programs for over two decades. Later scoring matrices (e. g. Henikoff and Henikoff, 1992, Jones et al. 1992b, Gonnet et al. 1992) are often very similar to them. It is generally assumed that these matrices reflect the relative log likelihood of substituting one amino acid for another in evolution (for a review and hierarchical classifications of scoring matrices, see May 1999).

To align less similar sequences, it is often necessary to introduce relative insertions and deletions to attain a maximum matching of amino acids. So alignment gap penalties, which can also be viewed as a relative log likelihood of deletion or insertion, should be introduced. The earliest gap penalty was a fixed one for each residue deleted or inserted, or a fixed penalty for a gap of any length (Needleman and Wunsch, 1970). The former often invoked a large number of short insertions or deletions while the latter one could lead to extremely long gaps. Both were not biologically ideal. The most common form of gap penalty used now is the affine gap penalty, which can be written as: g=a+bn, (where g is the applied penalty, a and b are the opening and extending parameters while n is the number of spaces in the gap. Often b is much closer to zero than a.) (Gotoh 1982, Altschul and Erickson 1986). It can be regarded as the generalisation of the first two classes of gap penalties. Algorithms for constructing optimal global or local pairwise alignments require O(mn) time with these gap penalty functions, where m and n are the lengths of the sequences been compared. ("O(mn)" here means the computing time of the algorithm is roughly proportional to the product of m and n.) More complicated gap costs have been defined (e.g. Miller and Myers 1988). For the class of "concave" gap penalties, we can still build optimal alignment algorithms that require only O(mn) time. However,

implementation of such algorithms is more complex and error-prone. Almost all popular alignment programs use affine gap penalties.

1.2.2 Pairwise sequence alignment

The simplest form of sequence alignment is the pairwise sequence alignment. Although the traditional way of manually making pairwise alignment with a paper and a pencil works well with short and very similar sequences, the volume and laboriousness of tasks soon went beyond human capability. To obtain the optimal alignment between two sequences, Needleman and Wunsch (1970) introduced the dynamic programming algorithm into bioinformatics. With an assumption that each residue substitutes independently, this algorithm finds a single optimal alignment path given an amino acid substitution scoring matrix and a gap penalty function. In this alignment, the most similar segments of two sequences are aligned while gap regions between them are minimised. Gotoh (1982) implemented a more efficient version. Smith and Waterman (1981) developed a slightly different algorithm. It detects the best alignment between subsequences of two sequences, which is often called local alignment, compared to the global alignment from the Needleman-Wunsch algorithm. Overall, dynamic programming algorithms are effective alignment methods. Alignments built using them are employed for different applications, especially for building multiple sequence alignments and phylogenetic trees. Nevertheless, as the computing time of these algorithms is roughly proportional to the product of the lengths of two sequences, they are not very fast algorithms compared to most heuristic database searching algorithms.

1.2.3 Heuristic searching algorithms

The dynamic programming algorithms are guaranteed to find an optimal alignment according to a specified scoring scheme. However, with the growth of sequence databanks, speed of these algorithms became an issue. For example, comparison of a sequence of average length against a typical sequence database may take a few hours on a standard PC nowadays. There have been many attempts to produce faster algorithms than dynamic programming. Heuristic searching algorithms are among the most successful ones. FASTA (Lipman and Pearson 1985, Pearson and Lipman 1988) first uses a fast technique to locate locally similar regions between two sequences, then rescores these regions with dynamic programming algorithm. BLAST (Basic Local Alignment Search Tool) (Altschul et al. 1990) approximates alignments that optimise a measure of local similarity based on well-defined mutation scores. It directly simulates the results that would be obtained by a dynamic programming algorithm. Both programs could miss the optimal alignments that are detectable with dynamic programming algorithm. In fact, the original implementation of BLAST finds ungapped alignments only. However, the missing of a small proportion of significant matches is compensated by the gain in speed. Database searching can take minutes or even seconds with these programs.

Altschul et al. (1997) developed the PSI-BLAST searcher to incorporate gapped alignment and iterative searching with PSSM (Position Specific Scoring Matrices, Henikoff and Henikoff 1994) into BLAST. The first round searching uses GAP-BLAST,

a new version of BLAST program, which produces gapped alignments of sequence pairs. From the detected significant hits, a PSSM is constructed and this is used to search the databank again. This process can be iterated until no further significant hits can be found. Although over-enthusiastic use of the iteration approach can give misleading results, PSI-BLAST has been shown to detect more remotely related sequences and generate alignments of better quality (e.g. Elofsson 2002). Because of its speed, accuracy in estimating alignment statistics, and friendly interfaces for users and programs, PSI-BLAST soon became the most popular program for sequence databank searching

1.2.4 Multiple alignment and hidden Markov model

It can be assumed that all proteins arose from a small number of common ancestors. However, given that most protein sequences have no detectable similarity with each other, describing evolutionary relationships between all protein sequences is a difficult task. Some clusters of sequences do share significant similarities according to alignment scores. Where similar functional and structural features and significant alignment scores imply the proteins have relatively recent common evolutionary origin, it is possible to classify protein sequences into families of homologous sequences. An alignment is a hypothetical model of mutations that arose during sequence evolution. For a protein family with more than two sequences, pairwise alignments between each sequence alone give an incomplete description of the evolution of the whole family. To build phylogenetic trees of sequences and hierarchical classifications of them, detect the conservation pattern of families, generate PSSMs for recognising family members,

predict protein secondary structure, find the correlated evolution between residues and for other proposes, pairwise alignments are combined into multiple sequence alignments.

The classic dynamic programming algorithm for pairwise sequence alignment could be generalised for more than two sequences. The MSA program by Lipman and co-workers (1989) uses this method to find multiple alignments. It is mathematically simple yet powerful, but slow and computationally demanding. Even with clever programming and careful implementation, the program is practicable for only about ten short protein sequences. We need faster methods for most multiple alignment tasks. Most automatic multiple alignment methods using the strategy called "progressive alignment" (Feng and Doolittle 1987, Taylor 1987). The multiple alignment is built up from a series of small, computationally inexpensive alignments. The pairwise alignment algorithm is generalised to construct alignments between an alignment and a sequence or two alignments. Larger and larger alignments are combined until all sequences are aligned together. Random order of alignment can be employed. However, most programs use a hierarchical classification and align the sequences according to the branching order in a guide tree. The most similar sequences are aligned first. The risk of introducing errors in these alignments is smaller. Then when more divergent sequences are to be aligned, the pattern of residue conservation could be interpreted from earlier alignments to help build more accurate alignments (Taylor 1988, Corpet 1988, Higgins and Sharp 1988). The progressive alignment programs are much faster than MSA and widely employed. New programs (e.g. Heringa 1999, Notredame et al. 2000) have been developed to improve performance within similar frameworks.

Generally, the best multiple alignment of a sequence family cannot be unambiguously established. New schemes have been proposed for weighting and scoring for multiple alignment (e.g. Gotoh 1995, Morgenstern et al. 1996, Notredame et al. 2000) and research is still ongoing. It implies a lack of a universally accepted framework for multiple alignment even though good multiple alignment programs are vital for many applications.

The HMM (Hidden Markov Model) has become a popular model for making and scoring multiple alignment in recent years (for a review, see Eddy 1998). The models for multiple alignment are closely related to the use of "profiles", and are often called profile-HMMs. Although HMMs have been used in many fields, profile-HMMs for multiple alignment are probably the most popular application of hidden Markov models in bioinformatics. Here a "profile" can be considered as a generalised model of a sequence of positions, each position having a score for each amino acid. In addition to the scores of amino acid matching, the profile-HMMs also specify the scores of insertion and deletion of residue segments. Given a multiple alignment of sequences from a protein family, a profile can be built (trained) while probabilities in this model are estimated according to the alignment. Once the profile of a family has been parameterised, we can align it with other sequences or obtain the multiple alignment of some sequences in this family via established algorithms. Profile-HMMs are more detailed descriptions of the residue conservation patterns of protein families than simple PSSMs.

1.2.5 Secondary structure prediction

HMMs have also been successfully employed in protein secondary structure prediction (Bystroff et al. 2000). Like methods such as the k-nearest neighbour method (e.g. Salamov and Solovyev 1997), it can predict secondary structure by local sequence similarity to segments of known structures. If a sequence segment has similar amino acid pattern with a segment of known structure, it is natural to assume that they share similar secondary structure compositions, since the formation of secondary structures is greatly influenced by the building of hydrogen bonds between local amino acids (Honig 1999).

Because the organisation of protein secondary structure is closely related to the composition of local sequence, tools from the pattern recognition field have been applied using local sequence conservation pattern to predict protein secondary structure. Possibly the most popular one is the artificial neural network model (ANN). Neural networks are organised as interconnected layers of neurons. Each neuron receives information from one or more other connected neurons and determines its output signal based on the weighted sum of input signals. With many neurons and weighted connections, a large neural network is capable of modelling extremely complex functions. The influential paper by Qian and Sejnowski (1988) proposed one of the earliest methods to use neural network for protein secondary structure prediction. Rost and Sander (1993, 1994) improved their work and developed the popular program PHD. It uses a series of neural networks to predict secondary structure of a probe sequence from its pre-generated multiple alignment. Using neural networks and residue exchange patterns extracted from

multiple alignments, PHD significantly improved the accuracy of prediction. It became probably the most widely employed secondary structure prediction method. In the later program PSIPRED by Jones (1999b), the multiple alignments are replaced by the profiles from PSI-BLAST searching. It is one of the most accurate programs tested in CASP (see later) experiments. However, these approaches utilise only local information. Programs (e.g. Frishman and Argos 1996, Baldi et al. 2000b) have been developed to incorporate long-range interactions for better prediction, especially for β strands with considerable success.

While secondary structure prediction programs can benefit from good multiple alignments, it is assumed that an accurate alignment of protein sequences from a family should have all (correctly predicted) corresponding secondary structures aligned together. So good secondary structure prediction should be able to help multiple alignment as well. Aiming to improve the accuracy of both, Heringa (1999) introduced a method to make iteratively multiple alignments and secondary structure predictions.

1.2.6 Comparative modelling

Comparative modelling of unknown protein structures is currently the most reliable method for protein structure prediction. This method is also frequently referred to as homology modelling. It is based on the observation that two proteins with very similar sequences tend to have similar (backbone) structures (Chothia and Lesk 1986). So, it can only be applied when there are protein structure templates that share clear sequence

similarity with the probe sequence. Once such templates are detected, we can predict the structure with comparative modelling methods. The procedure often includes building alignments between the templates and the probe sequence, copying the backbone structures from templates according to the alignments, building a framework structure for the probe, adding loops and side chains, refining and validation the model (e.g. Sutcliffe et al. 1987).

The quality of models from comparative modelling is clearly related to the similarity between templates and probes (Baker and Šali 2001). When the pairwise sequence identity between a probe and the template is higher than a certain threshold (e.g.60%), programs can build very accurate models (e.g. May and Blundell 1994, Moult et al. 1995). Comparative modelling methods are highly developed for such cases, producing models of quality comparable to structures from structure determination experiments. Even an automatic server is capable of generating good models (Peitsch 1996). With more remotely related template and probe, the building of loops and especially the alignment between the templates and the probe are still problematic. Protein fold recognition methods (see later) have been applied in comparative modelling to select structure templates and generate alignments between templates and probe sequences (e.g. Bates et al. 2001).

1.2.7 Fold recognition (threading)

The terms 'fold recognition' and 'threading' are sometimes confused. Fold recognition is a

technique to detect remote similarities between protein structure templates and probe sequences. Threading is a class of methods to perform fold recognition based on identification of stable residue contacts.

The basis of fold recognition is the fact that a large percentage of protein sequences adopt one of a limited number of folds (Orengo et al. 1997, Murzin et al. 1995). Although we can still obtain new folds every year from structure determination experiments, the number of new folds is relatively small compared to the number of folds we have observed (Orengo et al. 2001). For a probe protein sequence with unknown structure, it is likely that its fold has been seen and proteins with similar structures are available in structural databases. If the sequence similarity between the probe and a template is higher than a certain threshold (Sander and Schneider 1991, Abagyan and Batalov 1997), we can confidently assign them as homologues and predict the structure of the probe with comparative modelling methods. However, while protein structures are more conserved than sequences (Chothia and Lesk 1986), proteins with the same fold often have no detectable sequence similarity (e.g. Rost 1999). Fold recognition methods are designed to detect such similarities and generate alignments.

Fold recognition methods fall broadly into two categories: one performs 3D-1D matching, the other uses pairwise interaction potentials and is often called 'threading'.

The first fold recognition and 3D-1D matching method was developed by Bowie et al. (1991). By describing the structural environment of each residue in structure templates,

they attempted to match templates with sequences using the preference of amino acids in different environments. The environment was described in terms of local secondary structure, solvent exposure and the degree of burial by polar rather than apolar atoms. It was assumed that the residue structural environment is more conserved than the residue itself, so the method can detect more remote relationships than pure sequence based methods. The method has been improved by many researches (e.g. Rost 1995, Russell et al. 1996, Rice and Eisenberg 1997). Because of the improvements in secondary structure prediction accuracy, the predicted secondary structure and residue exposures of probe sequences were also included into the scoring scheme.

Broadly, many methods could be included in the 3D-1D matching category. The methods describe the fold (family) specific conservation patterns of residues with PSSMs (Sometimes profile-HMM methods are also included.). Information can be derived from structural environment of template residues (e.g. Johnson et al. 1993, Shi et al. 2001), structural and sequence alignments of templates (e.g. Jones 1999a, Kelley et al. 2000), and the sequence alignments of the probes from sequence database searching. All these programs use some forms of dynamic programming algorithm to generate alignments. They are more accurate in detecting remote relationships between templates and probes than simple sequence alignment algorithm, and often much faster than threading programs.

Jones et al (1992a) coined the term 'threading'. In their method, a given protein fold is modelled as a network of pairwise interactions between residues. A sequence is matched

to a structure by considering pairwise interactions, rather than local residue structural environments only. By including non-local interactions, threading programs aim to detect even more remote relationships between templates and probes. However, the inclusion of non-local interactions prohibits use of the classic dynamic programming algorithm, because the assumption of independence in dynamic programming algorithm is no longer valid. In the first threading program, an iterative approach, which was developed for protein structure alignment (Taylor and Orengo 1989) was introduced for making structure-sequence alignments. Recursive dynamic programming (Thiele et al. 1999), Gibbs sampling algorithm (Bryant 1996), and other heuristic algorithms (e.g. Huber and Torda 1999, Xu and Xu 2000) have been developed to generate alignments in more efficient ways.

Both comparative modelling and fold recognition methods require appropriate templates to be present in the structure library. When no template can be confidently identified, *ab initio* modelling methods can generate models without using full templates.

1.2.8 Ab initio modelling

Perhaps the most intuitive way of simulating protein folding is via molecular dynamic simulation with a physical potential function, because the physical interactions between atoms are clearly the driving force of protein folding. Obviously, we can predict protein structures via this approach without using structure templates. However, explicit representation of molecules and complex potential functions employed in such

approaches require huge computing power. Also, accurate modelling of potential functions is a challenging problem itself. Only groups with giant cluster of supercomputers like the IBM Blue Gene Project could be capable of performing such simulations for proteins of reasonable sizes.

With limited computing resources, most *ab initio* modelling methods work with greatly simplified models, which can be divided into two classes: lattice (e.g. Skolnick and Kolinski 1991) and off-lattice models (e.g. Park and Levitt 1995). Using these models can sufficiently reduce the complexity of the conformational search because many details of protein 3D structures, including coordinates of most atoms, are ignored. Once the representation of protein structures is specified, a scoring function must be developed to measure the quality of different predicted models. Physical potential functions are not feasible with these reduced complexity representations. Many methods utilise scoring functions derived from the protein structure database that were adjusted to favour the native conformation over others. Interestingly, these so-called knowledge-based pseudo-energy potentials (for reviews, see Sippl 1995, Jones and Thornton 1996, Moult 1997, Lazaridis and Karplus 2000) are often employed in threading programs as well. With simplified representations and scoring functions, *ab initio* modelling programs search for near-native structures with Monte Carlo (e.g. Simons et al. 1997) or other algorithms (Aszodi and Taylor 1996).

In spite of encouraging recent improvements (Simons et al. 1999, Bonneau et al. 2001), most *ab initio* modelling methods are still limited to short protein sequences. Also, to

build accurate models with *ab initio* methods remains a challenge.

1.2.9 CASP: the critical assessment of structure prediction

CASP (Critical Assessment of Protein Structure Prediction Methods) (Moult et al. 2001) is probably the most important experiment in protein structure prediction. Every two years since 1994, the organisers of CASP collect protein sequences from X-ray crystallographers and NMR spectroscopists while structures of these proteins are been solving or just solved. These sequences (called prediction targets) are made available to predictors through a web interface. Participants submit their predictions via email or web interfaces. Experimental structures of these proteins will not be published before the deadline of model submission. So, participators should have no access to the correct answers in predicting. Their models will then be manually and automatically evaluated via many quality measures (e.g. Zemla et al. 2001, Cristobal et al. 2001). The experiment addresses the capability of current methods of protein structure prediction. Categories of prediction include comparative modelling, fold recognition and *ab initio* modelling. Progress is published in special issues of the journal PROTEINS: Structure, Function, and Genetics. CASP is a community-wide blind test designed for critical assessment of protein structure prediction methods.

In the CASP protocol, human intervention is allowed in making predictions. Manual examination and modification of alignments in comparative modelling and fold recognition is widely employed by the most successful groups (e.g. Bates 2001, Murzin

and Bateman 2001). However, for biologists outside the field, experience of performing such tasks is hard to obtain. Also, manual examination of prediction results is clearly not feasible or reproducible in large- scale experiments. Automatic programs with convenient web interfaces are tested in the CAFASP (Critical Assessment of Fully Automated Structure Prediction experiment) (Fischer et al. 2000), which was initiated by Fischer and co-workers (1999). The results of CAFASPs and CASPs show that, in most cases, human intervention leads to better predictions. However, several programs can already independently produce reasonable predictions.

Chapter 2.

Amino acid substitution matrices from an artificial neural network model

An amino acid substitution matrix specifies probabilities of substitutions for each pair of the 20 amino acids. Log-odds scores transformed from the values in substitution matrices are widely used to construct protein sequence alignments. Any given substitution matrix is suited to matching sequences diverged by a specific evolutionary distance. However, for a given set of sequences, it is not always clear what matrix should be used. I used an artificial neural network model to predict probabilities of amino acid substitutions with alignment samples of different evolutionary distances. From this internal description, substitution matrices suitable for detecting relationships at any chosen evolutionary distance can be instantly generated. By using the additional information of evolutionary distances, the average cross entropy error of my neural network model is lower than that of a series of BLOSUM and PET matrices over all testing sets. My model is more accurate on the prediction of amino acid substitution probabilities.

2.1 Introduction

Most protein sequence analysis tasks rely on measures of similarity between different amino acids, typically encoded in amino acid substitution matrices. These matrices, usually derived from observed residue exchanges within protein alignments, specify the probabilities of substitutions between amino acids at different evolutionary distances. Log-odds scores from substitution matrices are widely used in dynamic programming-based protein sequence comparison (Needleman and Wunsch 1970, Smith and Waterman 1981) and database search programs (Pearson and Lipman 1988, Altschul et al. 1990).

Dayhoff and co-workers (Dayhoff et al. 1978) introduced a Markov model of evolutionary change in proteins. Here it was assumed that each mutational event was independent of previous events. The 20*20 mutation probabilities of amino acids at an evolutionary distance of 1 PAM (Point Accepted Mutation) were calculated using alignments of highly similar protein sequences, and substitution matrices for greater evolutionary distances were then extrapolated by repeated multiplication of the 1 PAM matrix. From these substitution matrices, log-odds matrices were generated for the scoring of protein sequence alignment. PETs (Pairwise Exchange Tables) are a series of substitution matrices generated using a similar approach and a larger data set (Jones et al. 1992b). Such matrices are still widely applied in many protein alignment programs.

The assumption of a Markov model in the PAM matrix formulation was questioned by

Benner et al. (1994) who found that PAM type substitution matrices for large evolutionary distances (built using only alignments of highly similar protein sequences) differ from those derived from alignments of more distantly related sequences. They further concluded that alignments of more distantly related proteins are required for construction of sensitive substitution matrices at large evolutionary distances. Similarly, Henikoff and Henikoff (1992) used distant sequence relationships but restricted their attention to the more conserved parts of multiple protein sequence alignments. Again, they calculated a series of substitution matrices (BLOSUM: BLOcks SUbstitution Matrix) to be applied at differing degrees of evolutionary divergence.

Three-dimensional structures are more conserved through evolution than their amino acid sequences (Chothia and Lesk 1986). It is for this reason that structure alignments are often regarded as the "standard of truth" for protein relationships (for example, see Brenner et al. 1998). This has led to the construction of substitution matrices based on structure alignments of proteins (Risler 1988, Johnson and Overington 1993). Alignments of analogue and homologue proteins have also been used to build different substitution matrices (Russell et al. 1997). Nevertheless, because of the limited number of available protein structures, if these alignments are divided into too many sub sets, the statistical significance of matrices would be affected due to lack of data.

It has been shown that different matrices are suitable for different alignment tasks and a given PAM matrix is best at finding segments that have diverged by a certain range of evolutionary distances (Altschul 1991, Henikoff and Henikoff 1993, McClure et al.

1994). To overcome this, Altschul built an all-range scoring matrix based on the PAM model. This matrix allows the evolutionary distance inherent in the similarities sought to be partly ignored at the cost of lost statistical significance (Altschul 1993).

Here I re-examine the generation of a universal scoring-scheme for amino acid substitutions using a neural network model. The including a measure of evolutionary distance improves the performance of this model and helps maintain an optimal scoring-scheme over a continuous range of evolutionary distance.

2.2 Methods

2.2.1 Data sets

The protein structure classification CATH (Orengo 1997) (v1.6) was used for selection of training and test sets. I used only domains with no break in the alpha carbon backbone. Firstly, 681 pairs of protein domains were selected, in which the two protein domains of each pair are in the same sequence family but not near identical structures. According to CATH, they have sequence identities greater than 35% (with at least 60% of the larger domain equivalent to the smaller), indicating highly similar structures. Then, another 339 pairs were selected, with each protein sharing the same homologue family with the other one in the pair, but being in different sequence families. All protein structural alignments were made using SAP (Structure Alignment Program) (Taylor and Orengo 1989, Taylor 1999). In SAP, the pairwise relationships are scored on the spatial position of residues relative to the local co-ordinate frame. This score ranges from 0 to several hundreds and most significantly similar residues score more than 1. To avoid noise from amino acids aligned without significant similarity, I set a threshold of SAP score at 1, discarding 8% of all aligned residue pairs with lower scores, and used only those with higher alignment scores. Four fifths of these alignments (800 randomly selected alignments, 186468 aligned residue pairs) were used for training and cross-validation of the neural network, the remaining fifth (220 alignments, 50758 aligned residue pairs) for testing.

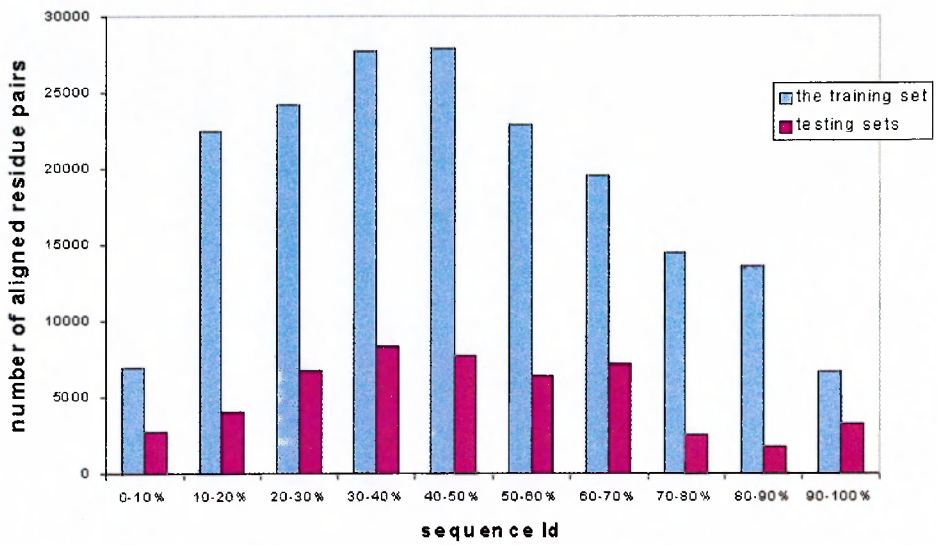


Figure 2.1. Distribution of testing and training sets over alignment sequence identity.

2.2.2 Training and testing of neural networks

The neural network has a three-layer feed-forward architecture with 22 input units, 10 hidden units, 20 output units and full connections. I used the standard logistic activation function

$$g(x)=1/(1+\exp(-x)) \quad (1)$$

for the hidden layer and the softmax activation function for the output layer (Bishop, 1995). Various network architectures were tested: the number of hidden units ranging

from 3 to 20. Most models with more than 6 hidden units performed well. The 10-hidden-units model, which achieved a low training error, was selected for further work. For each aligned residue pair in the training set (or testing set), length of alignment, sequence identity of alignment (both according to SAP) and amino acid type of one residue were used as the input. The amino acid type of the other residue was employed as the target. After the propagation of the neural network, the values of the output units Oi are interpreted as the corresponding predicted substitution probabilities of the amino acid presented as input (Figure 2.2).

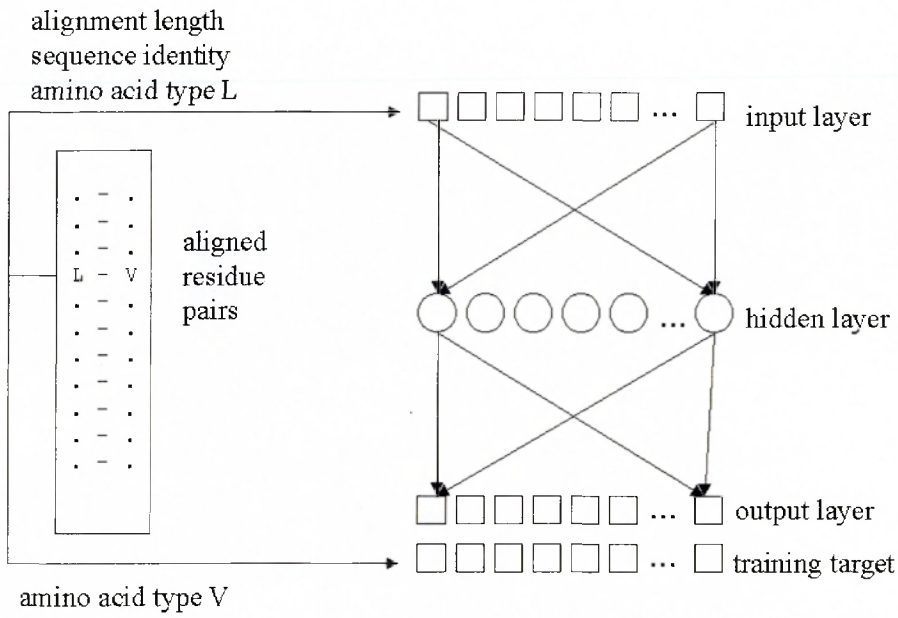


Figure 2.2. Training and testing of the neural network.
A pair of aligned amino acids was used here. Together with the presented alignment length and the sequence identity (according to the structure alignment), the Leucine was used for input, and the Valine for target. The differences between output and target were used for the calculation of cross-entropy error.

I used the online back-propagation algorithm (Bishop, 1995) to train the network. (Parameters of training algorithm: the momentum rate=0.5 and the learning rate=0.005.) Training was stopped using a six-fold cross-validation approach. After training, the performance of the neural network was tested using the test set and the average cross entropy error E was calculated.

It took 22 seconds to propagate this neural network model 1,000,000 times on a PC with

a Pentium III 500 chip. The compiler was GNU g++ 2.95. The operating system was Linux 2.2.

2.3 Results and discussion

For the trained neural network, I presented an amino acid to the input layer, propagated the net and collected all 20 predicted substitution probabilities corresponding to this amino acid from the output layer. I applied this operation to all 20 amino acids providing a full (20*20) substitution matrix, in which the rows provide probabilities of substitutions of each amino acid (Table 2.1).

Table 2.1 The substitution matrix from the neural network (presented alignment length 150 , sequence identity 50%)

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	0.441	0.030	0.020	0.033	0.000	0.029	0.045	0.064	0.011	0.021	0.043	0.032	0.010	0.015	0.029	0.069	0.041	0.003	0.009	0.055
R	0.034	0.519	0.026	0.012	0.001	0.047	0.033	0.022	0.014	0.020	0.029	0.134	0.007	0.005	0.019	0.041	0.013	0.004	0.015	0.004
N	0.029	0.036	0.396	0.097	0.001	0.034	0.048	0.051	0.015	0.014	0.021	0.053	0.006	0.009	0.009	0.088	0.054	0.002	0.017	0.025
D	0.039	0.012	0.070	0.522	0.000	0.031	0.102	0.042	0.008	0.003	0.006	0.036	0.002	0.007	0.014	0.047	0.031	0.002	0.011	0.016
C	0.035	0.012	0.007	0.006	0.757	0.007	0.003	0.012	0.007	0.021	0.031	0.008	0.003	0.009	0.004	0.023	0.023	0.001	0.015	0.019
Q	0.055	0.061	0.031	0.043	0.001	0.395	0.121	0.022	0.022	0.014	0.011	0.092	0.013	0.005	0.013	0.036	0.044	0.004	0.006	0.014
E	0.056	0.029	0.026	0.091	0.000	0.072	0.474	0.021	0.010	0.009	0.015	0.065	0.008	0.004	0.018	0.029	0.036	0.000	0.018	0.021
G	0.060	0.023	0.027	0.036	0.001	0.016	0.007	0.691	0.008	0.007	0.017	0.022	0.000	0.001	0.017	0.038	0.015	0.000	0.002	0.012
H	0.019	0.038	0.041	0.039	0.000	0.044	0.040	0.012	0.572	0.005	0.036	0.044	0.006	0.014	0.011	0.006	0.011	0.001	0.047	0.015
I	0.024	0.011	0.006	0.003	0.000	0.009	0.010	0.011	0.003	0.464	0.157	0.015	0.028	0.016	0.011	0.008	0.039	0.001	0.009	0.175
L	0.027	0.011	0.011	0.003	0.007	0.002	0.008	0.006	0.004	0.096	0.603	0.011	0.042	0.039	0.013	0.010	0.017	0.002	0.011	0.077
K	0.050	0.107	0.034	0.039	0.001	0.057	0.070	0.019	0.013	0.018	0.019	0.411	0.004	0.008	0.022	0.044	0.057	0.002	0.010	0.019
M	0.020	0.019	0.013	0.005	0.000	0.005	0.023	0.020	0.012	0.092	0.243	0.031	0.337	0.037	0.016	0.017	0.023	0.001	0.014	0.073
F	0.027	0.005	0.004	0.010	0.003	0.003	0.001	0.010	0.011	0.009	0.088	0.011	0.015	0.605	0.009	0.014	0.023	0.017	0.111	0.024
P	0.056	0.016	0.004	0.025	0.000	0.013	0.038	0.020	0.010	0.009	0.009	0.024	0.004	0.009	0.688	0.043	0.003	0.000	0.004	0.024
S	0.104	0.030	0.052	0.033	0.001	0.027	0.043	0.039	0.004	0.020	0.015	0.032	0.004	0.012	0.027	0.408	0.111	0.002	0.017	0.022
T	0.065	0.016	0.036	0.028	0.000	0.024	0.035	0.012	0.004	0.039	0.032	0.053	0.009	0.018	0.008	0.126	0.424	0.002	0.015	0.056
W	0.018	0.010	0.014	0.014	0.000	0.014	0.016	0.008	0.005	0.015	0.031	0.018	0.011	0.037	0.000	0.019	0.011	0.688	0.049	0.022
Y	0.027	0.012	0.010	0.018	0.000	0.007	0.011	0.020	0.028	0.004	0.028	0.004	0.010	0.128	0.003	0.021	0.024	0.000	0.635	0.011
V	0.056	0.004	0.015	0.011	0.007	0.006	0.017	0.007	0.005	0.146	0.112	0.018	0.017	0.022	0.015	0.023	0.048	0.003	0.007	0.463

The correlation coefficients r between this matrix and the PET substitution matrices for

different PAM distances are shown in figure 2.3.

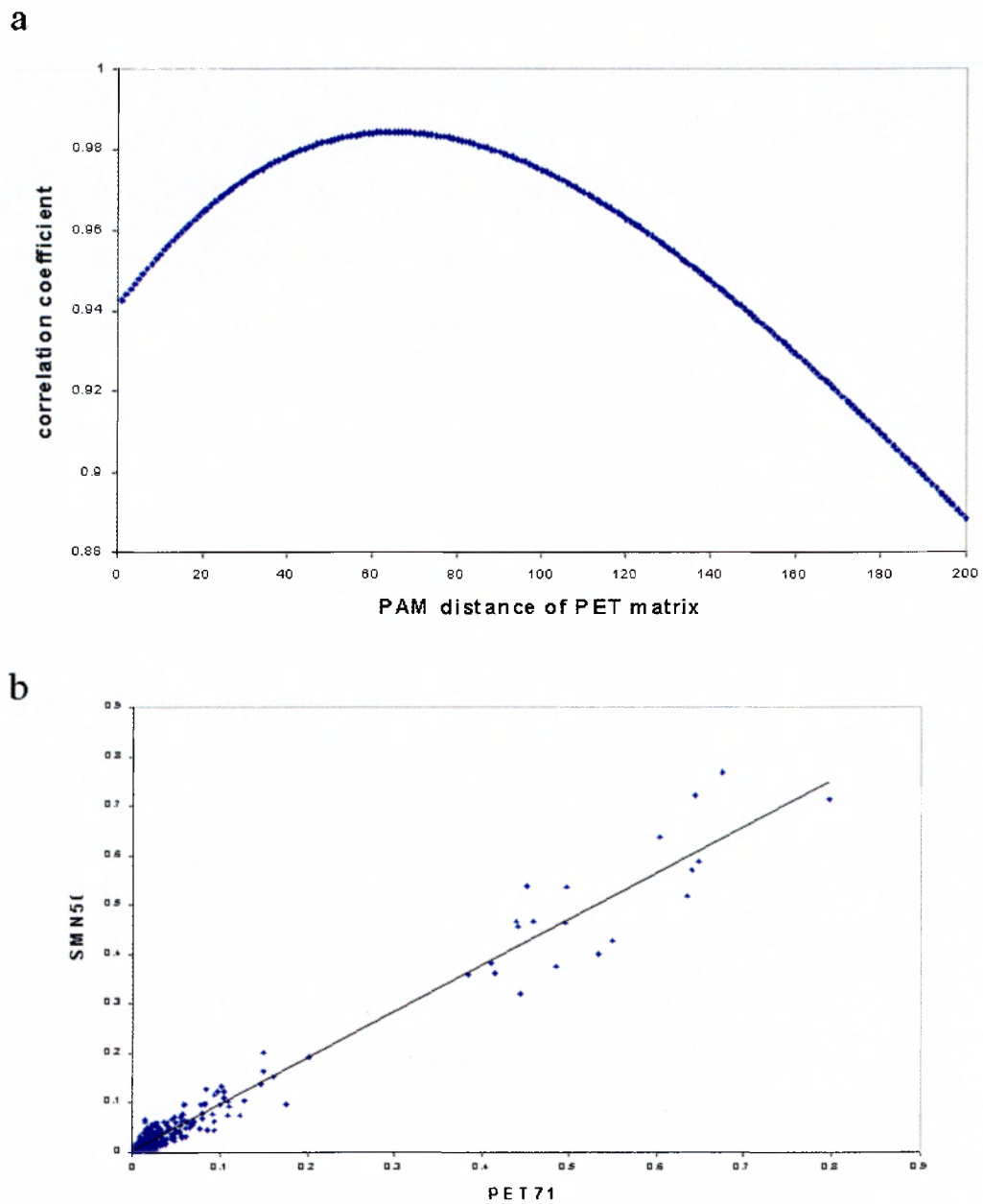


Figure 2.3 Comparison between the PET matrices and a SMN matrix.
a. Correlation coefficient between PET matrices and SMN matrix shows that they are highly similar. b. Comparison between values in PET71 and SMN50 matrices. The alignment length of this SMN matrix was set to 150 and the alignment sequence identity to 50%.

As expected, the behaviour of this neural network model is strongly affected by alignment sequence identity presented to its input layer. Figure 2.4 shows the shift of PAM distance of the most similar PET matrix (according to correlation coefficient) of this neural network model caused by the changing of presented alignment sequence identity. The SMN9 is most similar to PET250, with a correlation coefficient of 0.8460. The most different values between these two matrices are the probabilities of conservation of Tryptophan. According to PET, at the evolutionary distance of 250PAM, 46% of Tryptophan residues should be conserved while in SMN9 (Substitution Matrix from Neural network model, presented alignment sequence identity at 9%) this value is only 15%. Considering this value is 20% in BLOSUM30, I think SMN9 made a safer prediction. Because of the extrapolation approach in the generation of PET matrices, small amounts of noise introduced in PET1 will be enlarged to substantial level in PET250. Because Tryptophan has the smallest relative mutability (Dayhoff et al., 1978) and a low probability of occurrence, accurate estimation of mutation probabilities is more difficult for this amino acid. For larger evolutionary distances, I suggest my approach can make better estimations.

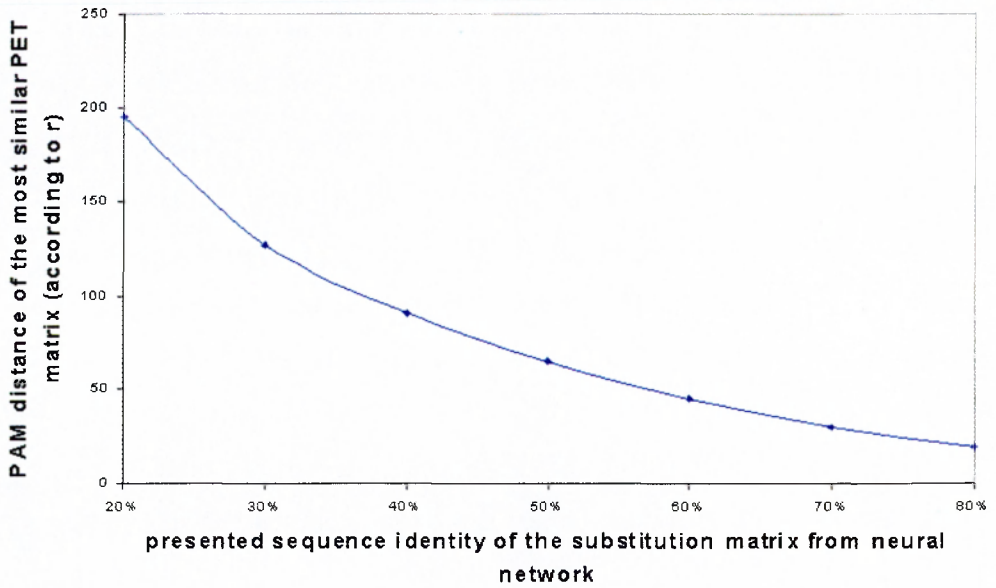


Figure 2.4. Relationship between PAM distance of the most similar PET matrix (from figure 2.3) and the alignment sequence identity set in SMN matrix.

Not only sequence identity but also length of alignment affects the statistical significance of alignments (Sander and Schneider 1991). In our scoring system, because of the flexibility of the neural network model, length and sequence identity of alignment are combined as a measure of evolutionary distance. However, changing presented alignment length did not have a marked effect on the generation of matrices. For SMN50, when the alignment length was changed from 150 to 50, the PAM distance of the most similar PET

matrix shifted from 71 to 69. The correlation coefficient between PET71 and SMN50 is still as high as 0.9830.

Altschul (1991) used the relative entropy H of a substitution matrix to measure the average information available per position to distinguish the alignment from chance. Here it was calculated using the same formula.

$$H=\sum_{ij}q_{ij}\ln(q_{ij}/p_ip_j) \quad (2)$$

The relative entropy of my matrices increases nearly linearly with presented sequence identities. This change is similar to that of various BLOSUM matrices, which increases nearly linearly with increasing percentage clustering (Henikoff and Henikoff 1992) (Figure 2.5). PAM30 has a relative entropy comparable to that of SMN80, PAM80 to SMN50 and PAM240 to SMN20.

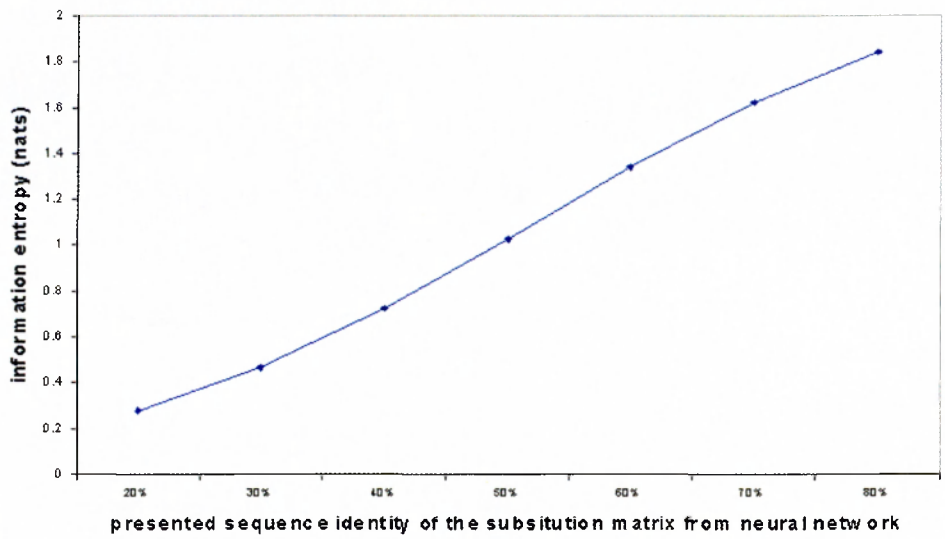


Figure 2.5. Relationship between information entropy and sequence identity set in SMN matrix.

Evaluation of substitution matrix performance using protein alignment programs can be greatly affected by the choice of gap penalties and alignment algorithms. To avoid this problem, here I treated all matrices and my neural network model as predictors in a multi-way classification problem with 20 categories that stand for the 20 different amino acids.

In this classification problem, I would like to model the posterior probability of class

membership in Bayesian inference, the probability of class k (the amino acid substituted by an amino acid with type k) given the feature x (the amino acid, alignment percent identity and alignment length) $p(C_k|x)$. However, I do not estimate the prior probabilities P(C) and class-conditional densities p(x|C) independently from data sets. I directly estimate posterior probabilities using a neural network model. For this purpose, I used the softmax (normalised exponential) activation function

$$y_k=\exp(a_k)/\sum_{k'}\exp(a_{k'}) \quad (3)$$

at the output layer of my neural network where a_k is the value of the output unit k before transition. Output value of each output unit was normalised so that outputs sum to 1, which is required for the output to be interpreted as posterior probabilities.

Assuming that each data point of input x and target t is drawn independently from the same distribution, the likelihood of observing this data set is then given by

$$L=\prod_n\prod_c(y)^t \quad (4)$$

the production over all n data point and c classes. For analytical simplicity I used the negative logarithm of the likelihood which leads to the cross-entropy error function of the form

$$E=-\sum_n\sum_c t_c^n\ln(y_c^n/t_c^n) \quad (5)$$

So, instead of maximising the likelihood, I used the back-propagation algorithm to minimise this cross-entropy error. If, for any n in test set, $t^n\equiv y^n$, then E=0 (Bishop 1995). However, here I use the size N of the test set to normalise this value so that results from different sets are comparable.

If I set the output of a predictor to the distribution of amino acid types regardless of the input, the average cross-entropy error of this predictor is the information entropy of this distribution. In this case, it is 2.91 *nats*.

The test sets were divided into 10 classes by sequence identities of alignments: 0-10%, 10-20%... and 90-100%, and the substitution probability matrices of BLOSUM, PET and SMN were tested on them (Figure 2.6, 2.7). As expected, BLOSUM40 and PET200 performed better over sets with low sequence identities, while BLOSUM80 and PET20 gave better results on sets from alignments with higher sequence identities. However, by using the additional information of evolutionary distances, SMN performed better over all test sets.

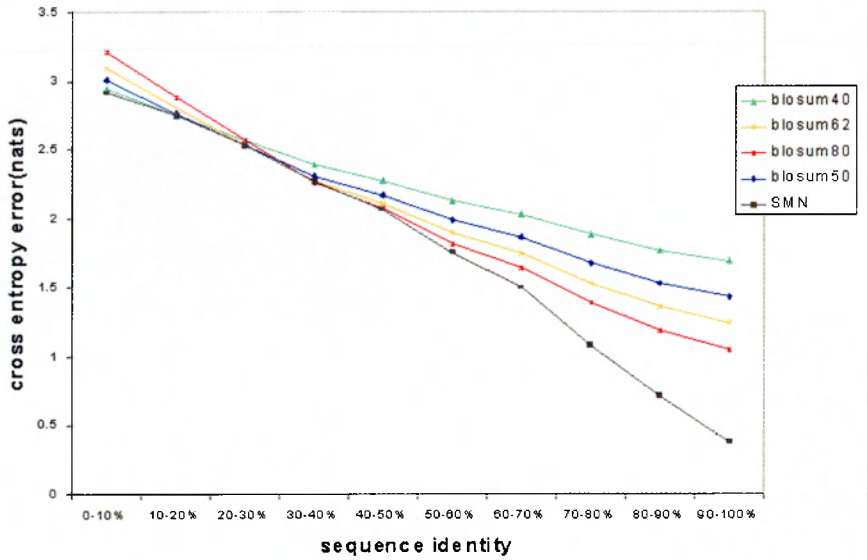


Figure 2.6. Test error (cross-entropy error) of BLOSUMs and SMN over testing sets of different alignment sequence identities.

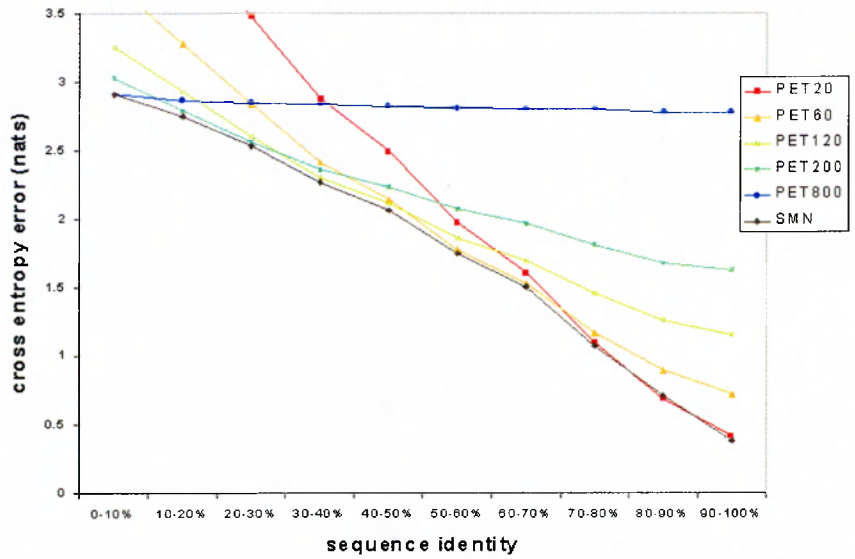


Figure 2.7. Test error (cross-entropy error) of PETs and SMN over testing sets of different alignment sequence identities.

2.4 Conclusions

Because of the ability of the neural network to recognise and generalise the relationships of amino acids at different evolutionary distances, I am able to use larger training sets. Mean alignment sequence identity of my training sets is 46%, the standard variance 0.24 (Figure 2.1). Directly constructing substitution matrices from this data set without consideration of evolutionary distance will lead to a matrix similar to SMN62. With two more input units, this neural network can simulate substitution matrices of very different evolutionary distances and provide a better performance on the prediction.

With my neural network model, substitution matrices of different evolutionary distances can be generated and used like other substitution matrices. However, this model can also be integrated with alignment programs, so that it can be instantly adjusted to changing conditions. This could be of particular importance in multiple sequence alignment programs where the optimal choice for each pair of sub-alignments must be made without any user intervention. In this situation, the matrix can also be adjusted according to the result of a test alignment, and used for generating a re-alignment.

Chapter 3.

Amino Acid Encoding Schemes from Protein Structure Alignments: Multi-dimensional Vectors to Describe Residue Types

Bioinformatic software has used various numerical encoding schemes to describe amino acid sequences. Orthogonal encoding, employing 20 numbers to describe the amino acid type of one protein residue, is often used with artificial neural network (ANN) models. However, this can increase the model complexity, thus leading to difficulty in implementation and poor performance. Here I use ANNs to derive encoding schemes for the amino acid types from protein three-dimensional (3D) structure alignments. Each of the 20 amino acid types is characterized with a few real numbers. My schemes are tested on the simulation of amino acid substitution matrices. These simplified schemes outperform the orthogonal encoding on small data sets. Using one of these encoding schemes, I generate a colouring scheme for the amino acids in which comparable amino acids are in similar colours. I expect it to be useful for visual inspection and manual editing of protein multiple sequence alignments.

3.1 Introduction

The artificial neural network (ANN) is a sophisticated modeling technique capable of modeling extremely complex functions and automatically learning the structure of data (Bishop 1995). ANNs have been widely applied to many different problems in bioinformatics (for reviews, see Baldi and Brunak 1998, Wu and McLarty 2000).

In neural network methodology, samples are often subdivided into "training" and "testing" sets. The training set is a set of examples used for "learning": fitting the parameters (i.e., weights) of a neural network. The testing set is a distinct set of examples used to assess the performance of a trained neural network. It is important to maintain a strict separation of these data sets with the testing set being applied only after determination of network architecture and connection weights.

A basic assumption in neural network training (and model optimization approaches of other machine learning methods) is that the training data exhibits an underlying systematic aspect but is corrupted with random noise (Bishop 1995). The central goal of model optimization is to produce a system able to make good predictions for cases not in the training set. It requires the model to represent the underlying mechanism correctly. Training an over-complex model may fit the noise, not just the signal, leading to "overfitting". Such a model will have low training error but a much higher testing error. Generally, its performance on new cases will be poor. The best way to avoid overfitting

is to use a large and diverse training set. However, given a training set of a limited size, model selection can be employed to improve generalization. With small training and testing sets, simpler models are often preferable for better performance (Bishop 1995, Müller et al. 1996).

Orthogonal encoding of amino acid types has been used in many bioinformatic neural network models: 20 input units are assigned to describe one protein residue. In the 20-dimensional space, the vector [1,0,0,0 ... 0,0,0] represents Alanine, and [0,0,0 ... 0,0,0,1] stands for Valine. With this encoding, a typical input window of 13 residues requires 260 (13*20) input units. It can easily lead to large input layers, many connecting weights, and hence complex models. Without sufficient data to support training, over-complex models are prone to overfitting. Unfortunately, in many bioinformatic problems, huge data sets can be simply unavailable. Even when they are available, analyzing them is often very computationally demanding. Simplified encoding schemes use less input units to describe a given amino acid sequence, thus we can use smaller models to describe the same phenomena. By introducing these simplified models, we can reduce the reliance on huge data sets and improve performance. To increase the level of neural network generalization, Skolnick and co-workers (1997) defined a 10-unit input scheme for representation of amino acid type. Each amino acid was described using ten numbers. Their representation was based on the amino acid features described by Taylor (1986): each unit corresponds to one biochemical feature, amino acids sharing many features have similar codes. Weiss and Herzel (1998) suggested two differing properties, "sequence derived hydrophobictiy" and "sequence derived polarity", based on

correlations in protein sequences. Jagla and Schuchhardt (2000) applied an adaptive encoding neural network to find automatically a classifier with a low dimensional encoding matrix. Their encoding scheme was tested on the prediction of cleavage sites in human signal peptides of secretory proteins.

Here, I use a supervised back-propagation neural network model to develop a series of schemes using several (1 to 10) input units to describe an amino acid. In these low dimensional representations, amino acids with similar biophysical properties are clustered together. These schemes are tested on the simulation of amino acid substitution matrices. With small training sets, simpler schemes can achieve better results. By using those simplified encoding schemes, we can greatly speed up the propagation and training of neural network models.

There is a clear need for a well-grounded amino acid colouring scheme to ease the interpretation of sequence alignments. Colouring comparable amino acids in similar colours facilitates manual examination and modification of sequence alignments. Different approaches have been taken to colour amino acid types according to their hydrophobicity, size and other biochemical properties (for example, Taylor 1997b). Here I generate a Red-Green-Blue (RGB) colour scheme by linearly transforming the values in my encoding scheme with 3 hidden units. This automatically constructed scheme can be easily adapted for other bioinformatic software. I write a simple Java program to browse protein alignments with this colouring scheme.

3.2 Methods

3.2.1 Data sets

I use the CATH protein structural domain database (Orengo 1997) (v1.6) to select training and testing sets. Domains with breaks in their alpha carbon backbones are excluded. Firstly, I select 681 pairs of protein domains, in which the two domains of each pair are in the same sequence family but not near identical structures. Then, another 339 pairs are chosen, with each domain sharing the same homologue family with the other one in the pair, but being in different sequence families. I align these domain pairs using Structure Alignment Program (SAP) (Taylor and Orengo 1989, Taylor 1999). In SAP, the pairwise relationships between residues from different domains are scored on the spatial position of residues relative to the local co-ordinate frame. The score ranges from 0 to several hundreds and most significantly similar residue pairs score more than 1. Thus, to avoid noise from amino acids aligned without significant similarity, I set a threshold of SAP score to 1. Aligned residue pairs with lower scores are discarded (8% of all aligned residue pairs). I use four fifths of these structure alignments (800 randomly selected alignments, 133609 aligned residue pairs) for training of the neural network, the remaining fifth (220 alignments, 33825 aligned residue pairs) for testing.

3.2.2 Construction of the encoding schemes

I use a feed-forward neural network with the logistic transformation function

$$f(x)=1/(1+EXP(-x)) \quad (1)$$

I employ the back-propagation algorithm with the root-mean-squared (RMS) error function. A 6-fold cross-validation approach (Bishop 1995) is used in the training. Each model is randomly initialized and trained 10 times. Only the model with the lowest cross-validation error will be used for further analysis. Details of the training procedure were described in our previous paper (Lin et al. 2001).

After training this neural network, I present each amino acid type to the input layer, propagate the network, and take the values of the hidden units as the encoding of the according amino acid. Here the size of the hidden layer determines the size of encoding schemes.

The encoding scheme based on the recognition of human signal peptide cleavage sites is obtained from Jagla and Schuchhardt (2000).

3.2.3 Testing

I test different encoding schemes on the simulation of substitution matrices using the same cross-validation approach (Lin et. al. 2001). For each encoding scheme, I adjust the size of the neural network input layer, translate the input amino acid types to corresponding codes, and perform the same training procedure. Three training sets of different sizes are employed. However, all models are tested on the same testing set, even

when the test set is much larger than the training set (Figure 3.1).

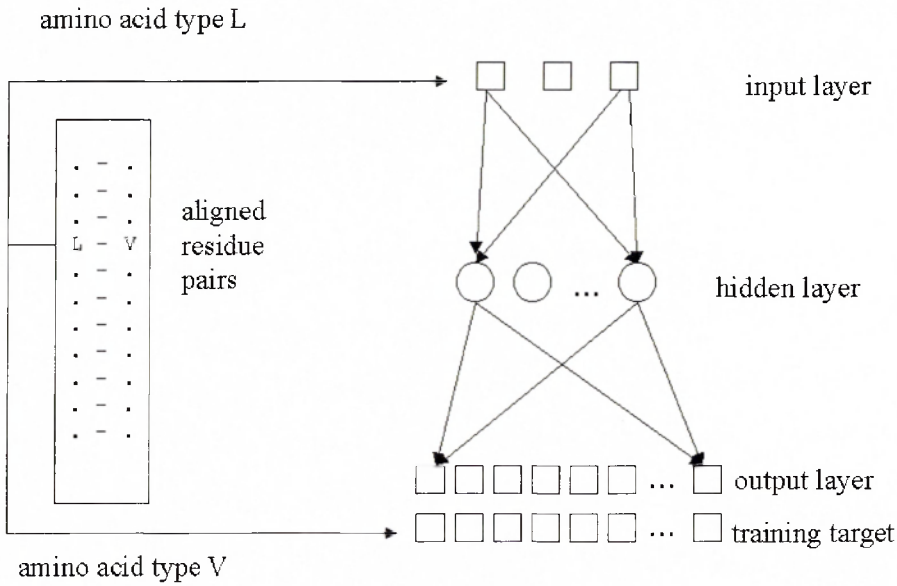


Figure 3.1. Testing of the encoding scheme. Aligned residue pairs are collected from structural alignments. For each pair, one residue is used as the input and the other as the target. Here the input amino acid Leucine is described using AESNN3 (Amino Acid Encoding Schemes from Neural Networks, size 3). The target amino acid valine is shown in the orthogonal encoding scheme. The output of this network is predicted substitution probabilities. This vector is compared with the target. The relative entropy between them is calculated (Bishop, 1995, Baldi et al. 2000a).

3.3 Results and Discussion

There are different approaches to measure the complexity of a neural network model (e.g. Maass, 1995). In the testing, the size of the hidden layer and output layer of models are specified. A larger encoding scheme directly leads to a larger input layer, more weighted connections and a more complex model.

An important feature of an encoding scheme is its size. Small encoding schemes use only a few numbers to describe types while large schemes employ many units. The simplest scheme tested here has a size of zero: the model completely ignores input amino acid types, and therefore the network only reflect the probabilities of amino acid types. The largest one, the orthogonal encoding scheme, utilises one input unit for each amino acid type. I have tested all intermediate sized schemes from my neural network models. Smaller schemes bring simpler models, which often perform better on small training sets. Nevertheless, models with two schemes of the same size can have different testing errors because of the different composition of the schemes. I want to find an approach to optimise automatically schemes so that small schemes can most efficiently describe the amino acid types.

With the largest training set, almost all models with size > 3 can achieve good generalization: Testing errors are low and differences between training and testing errors are small. Both the training and testing errors decrease with scheme size. With this set,

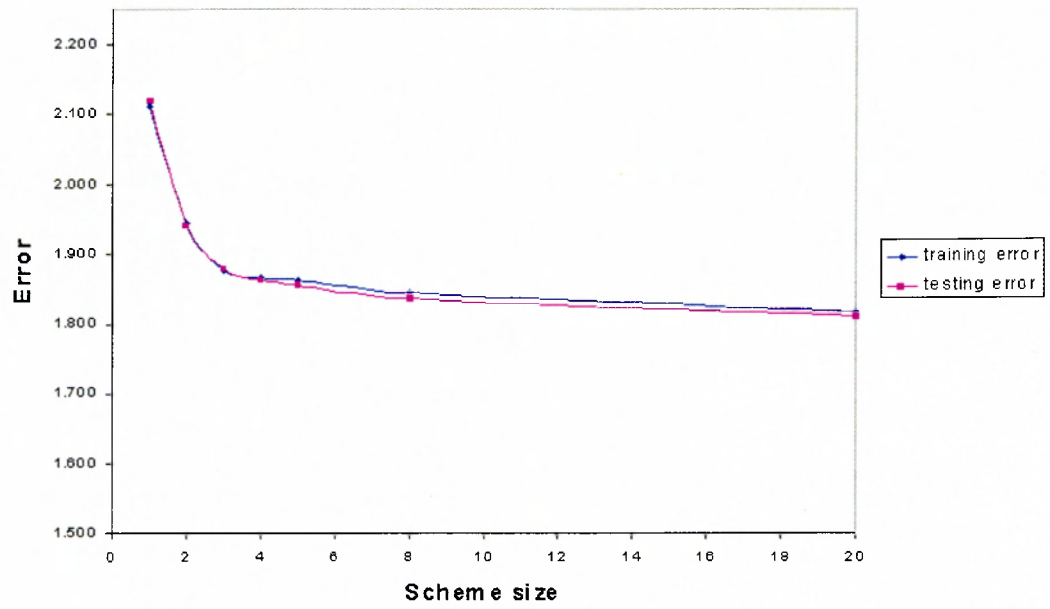
overfitting is far from problematic: the most complex model gives the best performance (Figure 3.2a). When I change to the smallest training set, all models are overfitted in training: Testing errors increase and training errors decrease. Small models like the model with amino acid encoding scheme from neural network, size 3 (AESNN3) (Table 3.1) and AESNN4 have the lowest testing errors (Figure 3.2b).

Table 3.1 AESNN3

A	-0.99	-0.61	0.00
R	0.28	-0.99	-0.22
N	0.77	-0.24	0.59
D	0.74	-0.72	-0.35
C	0.34	0.68	0.35
Q	0.12	-0.99	-0.99
E	0.59	-0.55	-0.99
G	-0.79	-0.99	0.10
H	0.08	-0.71	0.68
I	-0.77	0.67	-0.37
L	-0.92	0.31	-0.99
K	-0.63	0.25	0.50
M	-0.80	0.44	-0.71
F	0.87	0.65	-0.53
P	-0.99	-0.99	-0.99
S	0.99	0.40	0.37
T	0.42	0.21	0.97
W	-0.13	0.77	-0.90
Y	0.59	0.33	-0.99
V	-0.99	0.27	-0.52

Each amino acid type is described using a three-dimensional vector.
Values are taken from the 3 hidden units from the neural network trained on structure alignments
We linearly transform the values to the range (-1,1).

a



b

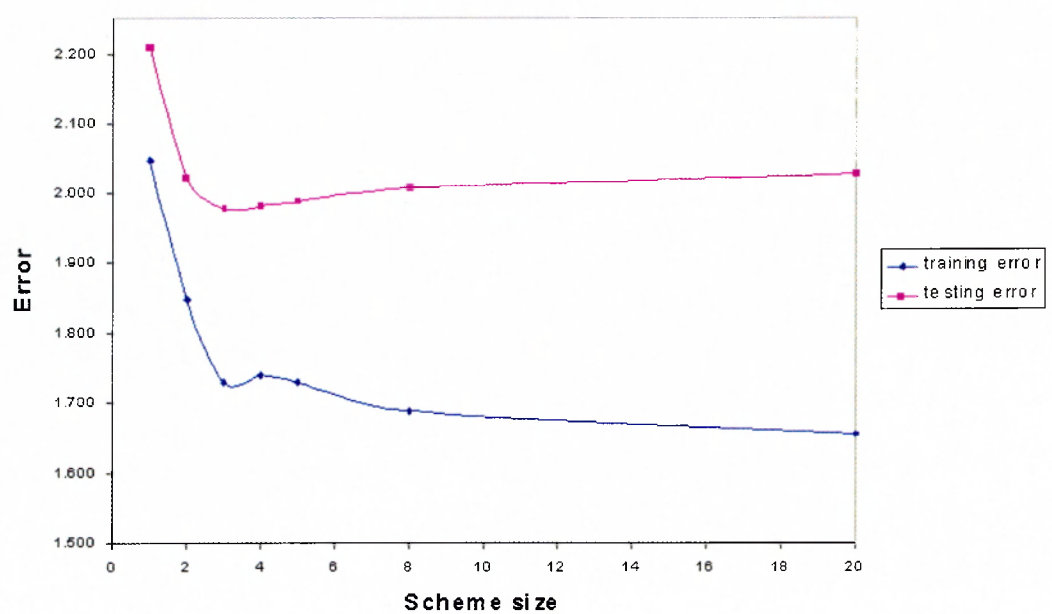


Figure 3.2. Training and testing errors of encoding schemes: (a) on the largest training sets (133609 aligned residue pairs). (b) on the smallest training sets (1350 aligned residue pairs). Test errors are from the same testing set (33825 aligned residue pairs).

The most complex model, which uses the orthogonal encoding scheme, has the largest difference between its training and testing errors. Although I used the cross-validation approach in the training, this model still suffers overfitting. On the other hand, schemes that are too small (like AESNN1 and AESNN2) are less than adequate for the description of amino acid types. Models with these small schemes perform badly on all training sets. AESNN3 and AESNN4 are recommended: on the largest training set, their testing errors are not much higher than that of the orthogonal encoding; with the smallest set, they are better (Table 3.2).

Table 3.2 Training and testing results of different schemes

		Training set 1		*Training set 2*		*Training set 3*	
Name	Size	Training	Testing	Training	Testing	Training	Testing
AESNN1	1	2.112	2.118	2.108	2.128	2.046	2.209
AESNN2	2	1.942	1.940	1.896	1.918	1.849	2.021
AESNN3	3	1.877	1.878	1.852	1.861	1.729	1.977
AESNN4	4	1.865	1.863	1.888	1.863	1.739	1.982
AESNN5	5	1.863	1.855	1.866	1.855	1.729	1.989
AESNN8	8	1.843	1.836	1.832	1.848	1.688	2.007
JNS	2	1.992	1.989	1.967	1.974	1.901	2.080
ORT	20	1.816	1.811	1.779	1.826	1.655	2.028
Size oftraining sets:		133609		13043		1350	

size:size of encoding schemes
training: cross-validation training error (relative entropy in *nats*)
testing: testing error (relative entropy in *nats*)
AESNN: encoding schemes from our neural network models
JNS: the encoding scheme from Jagla and Schuchhardt (2000)
ORT: the orthogonal encoding scheme
size of training sets: number of aligned residue pairs in training sets

Another reason to use small encoding schemes is speed. In training, the model with AESNN3 is about 9 times faster than the model with the orthogonal encoding (data not shown). To propagate a simpler model is slightly faster as well. It can be a considerable factor when we are dealing with huge sequence databases.

By analyzing 18 amino acid substitution matrices derived from different procedures, May (1999) gives a list of the reliable residue clusters after hierarchical classification of the 20

amino acids. Amino acids are grouped according to relationships confirmed by different matrices. All groupings of amino acids ranked more than 4 in this list (occurring more than 4 times in different classifications) are clustered to adjacent regions in my AESNN3. This observation strengthens the soundness of my projections.

Here I present a series of encoding schemes of the amino acid types. They can perform better than the traditional orthogonal encoding on small data sets in the simulation of an amino acid substitution matrix. It can be assumed that for different tasks of sequence analysis, different properties of residues are needed in descriptions (in this problem, AESNN2 performs slightly better than JNS2, the encoding scheme of Jagla and Schuchhardt, 2000). Using this approach, we can develop different encoding schemes optimized for prediction of protein secondary structure, prediction of contact matrices, etc. But I suggest these encoding schemes based on the simulation of substitution matrices can be used for general purposes.

Figure 3.3 shows my amino acid colouring scheme. In this colouring scheme, hydrophobic amino acids like methionine, leucine, isoleucine, valine, tryptophan, and phenylalanine are coloured in yellow-green colours. Polar amino acids are coloured in red, blue and purple. Proline is in black. My colour scheme is automatically constructed according to the evolutionary relationships between amino acids encoded in protein structure alignments. However, it confirms some features identified in previous work of Taylor (1997b) and May (1999). Without any arbitrary considerations, it should reflect more precisely properties of amino acids and their evolutionary relationships. I hope that

my colouring scheme will be useful for manual analysis of protein alignments. A simple Java program has been written to demonstrate colouring schemes described here and by Taylor (1997b). My encoding schemes, colouring scheme and this program are available on the web at http://mathbio.nimr.mrc.ac.uk/kxlin/aesnn/.

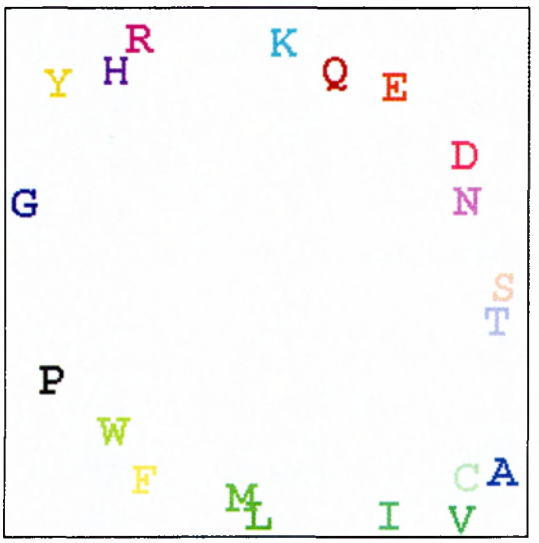


Figure 3.3. The colouring scheme.
Amino acid types are coloured according to AESNN3. For example, in AESNN3 Alanine is described as the vector [-0.99,-0.61,0.00]. I linearly transfer values in the scheme from the range (-1,1) to [0,255]. So, the letter A is coloured in RGB colour [1,49,126]. Two-dimensional co-ordinates of letters are from AESNN2.

Chapter 4.

Threading Using Neural Networks (TUNE): the measure of protein sequence-structure compatibility

Fold recognition programs align a probe protein sequence onto protein three-dimensional (3D) structure templates. The alignment between the probe sequence and the most suitable template can be used to predict the 3D structure and often biological function of the probe. Here I present a new threading scoring function of protein sequence-structure compatibility. An artificial neural network model is trained to predict compatibility of amino acid side-chains with structural environments. Log-odds scores of predicted probabilities from this model can then be used to construct protein sequence-structure alignments. My model was tested on the discrimination of native and decoy protein 3D structures. With a residue level structural description, its performance is comparable to those of pseudo-energy functions with atom level structural descriptions, better than the two functions with residue level structural descriptions.

The C++ source code of my neural network model is available at http://mathbio.nimr.mrc.ac.uk/~kxlin

4.1 Introduction

Fold recognition programs identify the optimal alignment between a probe protein sequence and the most suitable target protein 3D structures in a set of templates. With additional structural information from the templates, these programs are designed to detect remote relationships between proteins: those undetectable with sequence alignment programs. Normally, a threading program comprises two components: an alignment algorithm (the means of performing sequence-structure alignment) and a measure of protein sequence-structure compatibility. For the latter, pseudo-energy functions (also known as knowledge-based potentials, potentials of mean force or statistical effective energy) provide a measure of energy related to the probabilities of observing the proposed interactions in native protein structures (for reviews, see Sippl 1995, Jones and Thornton 1996, Lazaridis and Karplus 2000). These potentials have been widely employed in threading programs, comparative modelling, *ab initio* prediction of protein structures, simulation and prediction of ligand binding sites.

The efficiency of pseudo-energy functions depends strongly on the degree of detail of structural description. Pseudo-energy functions with detailed atomic structural descriptions have been shown to be more accurate than those with residue level structural descriptions (Samudrala and Moult 1998, Lu and Skolnick 2001). However, the aim of threading programs is to detect remote relationships between proteins. Sequence identities within alignments generated by them are often lower than 30%. Thus, even

when the probe sequence and the structure template share the same fold, most structurally equivalent residues can have different side-chains. Therefore, atomic knowledge-based pairwise potentials are not practical in these programs. Most pseudo-energy functions employed in threading programs simplify protein 3D structures as chains of interacting centres, using one or two co-ordinates to describe one residue, ignoring the co-ordinates of most atoms.

Although there are different approaches to build pseudo-energy functions, a common assumption is that pairwise contacts between atoms (or residues) have independent contributions to the pseudo-energy (compatibility). In my approach, I ignore the energy of contacts and try to predict the probabilities of observing amino acid side-chains in structural environments. I employ an ANN (artificial neural network), a non-linear mathematical model, to predict these probabilities. In this Bayesian framework, I use an integrated structure environment description to represent different structural features, including multiple contacts to one residue side-chain. This description is optimised using information theory (Shannon 1948). Information entropy is calculated to measure reduction in uncertainty in the prediction that results from including features. Features with less mutual information are considered less helpful to the prediction and thus removed from the description. Similar noise removal methods have been applied to many different bioinformatic problems (e.g. Bienkowska et al. 1999, Solis and Rackovsky 2000).

I test my model on the discrimination of protein decoy and native structures. Even with a

less detailed, residue level, structural description, its performance is comparable to pseudo-energy functions with atom level structural description. I hope that more accurate fold recognition methods can be developed with this model.

4.2 Data and methods

4.2.1 The description of structural environments

Each residue is described using two spheres: the sphere for the main-chain and the sphere for the side-chain (Figure 4.1). All spheres are considered to have the same density, and so, their radii are proportional to the cube roots of their mass. I extend the bond between the alpha and beta carbon to the radius of the side-chain, where the centre of the side-chain sphere is placed. The centre of the main-chain sphere is at the carbonyl carbon on the backbone. I do not use the alpha carbon here because it will lead to large overlap between the main-chain and the side-chain. Using this model, the space of a protein's core will be occupied one sphere only, more than one spheres or nothing. The sphere radii are optimised so that the volume of space occupied by multiple spheres or nothing is minimised, thus with the model, most space of a protein's core should be occupied by one sphere only. The radius of an Alanine side-chain sphere is 1.7 Å. The largest side-chain spheres have a radius of 3.5 Å. They are comparable to some published values (Cootes et al. 1998).

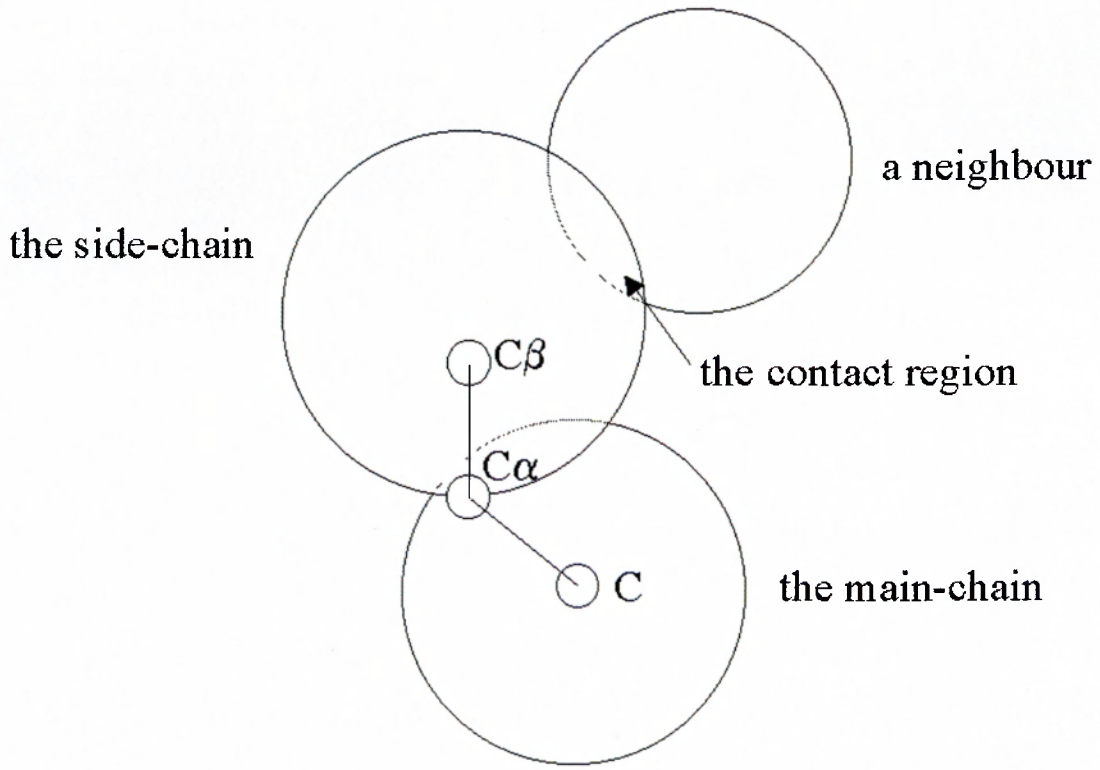


Figure 4.1. Description of structural environment
Each residue is described with two spheres: the sphere for main-chain atoms and the sphere for the side-chain. The centre of the main-chain is placed on the carbonyl carbon, and the centre of the side-chain on the pseudo beta carbon (see method). The contacts between the side-chain and its neighbours are encoded by the volume of their contact regions.

To simplify the description of contacts between a residue side-chain and its neighbours, main-chain neighbours are treated as side-chains of Asparagine (because of their similar chemical compositions), and the contact between the side-chain of a residue and its main-chain is ignored. First, I construct a default side-chain sphere. Second, I calculate the volumes of contact regions between this sphere and its neighbours. Since there are 20

different types of neighbours, 20 real numbers are used to describe different contact patterns. Each number is the sum of volumes of contact regions between the default side-chain and the neighbours of this specific type.

Of course, the radius of the default side-chain sphere greatly influences detection of contacts. With too small a radius, few contacts will be included into the description, while with too large a one, useful signals could be hidden by "irrelevant" contacts. I here apply an information theory approach to guide the selection of this value.

The information entropy of amino acid type of the central residue H(X) is defined by

$$H(X)= -\sum P(x)\ln(P(x)) \quad (1)$$

Where P(x) is the probability of observing amino acid type x. It measures the diversity of the distribution X of amino acid types of the central residue.

Here I define the feature Y as the two amino acid types that constitute the largest overlap in my description. They are often the amino acid types of the two closest neighbours.

The mutual information H(X;Y) is given by

$$H(X;Y)=H(X)-H(X|Y) \quad (2)$$

Where the conditional entropy H(X|Y) of the central residue given the feature Y is

$$H(X|Y)= -\sum P(x,y)\ln(P(x|y)) \quad (3)$$

The mutual information H(X;Y) measures the average reduction in uncertainty about X that results from learning the value of Y. It is least when X and Y are independent,

P(x,y)=P(x)P(y), and greatest when H(X|Y) or H(Y|X) is zero: which means X can be predicted with certainty given feature Y or *vice versa*.

Figure 4.2 shows the mutual information of X and Y when varying the radius of the default side-chain sphere. To improve the performance of the model, the radius is set to 4Å.

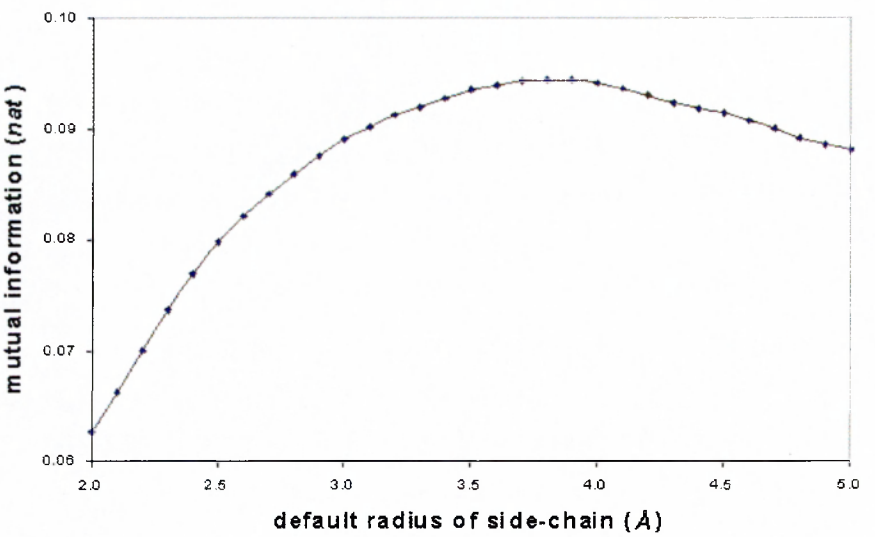


Figure 4.2. Optimisation of the contact description
The mutual information of the contact description is related to the default residue side-chain radius (see method). The mutual information measures the average reduction of uncertainty in prediction from including the feature. To maximise the mutual information, the radius is set at 4Å.

I use another five input units to describe other features of the structural environment. One represents residue exposure, measured by the sum of volumes of all contact regions. The other four are distances from the alpha carbon to the alpha carbons of +4, +2, -2 and -4 residues on the sequence.

4.2.2 Training of neural network models

The protein structure classification CATH (v2.0) (Orengo et al. 1997) is used to select training and test sets. From 2667 sequence family representatives, I obtained the structural environment description and the amino acid type for 412680 residues. All native structures in the decoy sets used for assessing ANN models, and domains in their sequence families, are kept for testing. Then, three fourths of all domains (2000 domains, 309168 residues) are randomly selected for training. The remaining domains (667 domains, 103512 residues) form the testing set, which is used only after construction of ANN models.

For each residue in the training set, its structural environment is described with 25 real numbers. The ANN model is trained to predict the probabilities of observing different amino acid types in this environment. A back-propagation algorithm is employed to minimise the mean difference between the predictions and real amino acid types. The training algorithm and parameters were described in our previous works (Lin et al. 2001, Lin et al. 2002a). Figure 4.3 shows the training of my ANN model.

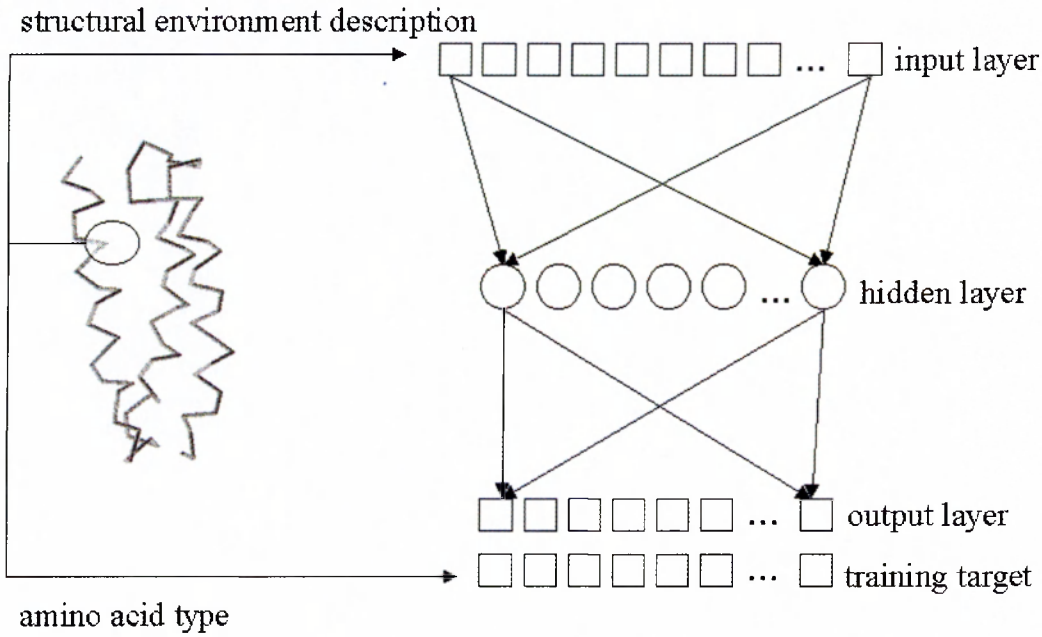


Figure 4.3. Training of the ANN model
For each residue in my training set, I describe its structural environment by 25 real numbers. I enter these real numbers into the input layer (25 units), propagate the signal through the hidden layer (22 units) to the output layer (20 units). The training target is set to its amino acid type, encoded by the orthogonal encoding scheme. The back-propagation training algorithm modifies the connection weights to minimise the difference between the output layer and the training target.

4.2.3 Testing of models

Given a residue in a protein structure, I encode its structural environment, enter the description into my trained ANN model and predict the posterior probability P(x|y): the probability of observing its amino acid type x given the structural environment y. The score of the compatibility is given by

$$S = \ln (P(x|y)/P(x)) \quad (4)$$

where P(x) is the occurrence of the amino acid type x (the probability of observing amino acid type x given no structural information). I assume that the higher this logarithm likelihood score is, the better this residue fits into its structural environment.

I take the three largest decoy sets from the Prostar website (http://prostar.umbi.umd.edu). For each structure, I sum the compatibility scores of every residue (side-chain). If the summed compatibility score of the native structure is higher than that of the decoy, I consider that my model performed correctly in the discrimination of this native-decoy pair.

The decoy sets from Decoys'R'us (http://dd.stanford.edu) (Park and Levitt 1996) are generated using a four-state off-lattice model together with a relaxation method. I evaluate all decoy and native structures with my ANN model.

4.3 Results and Discussion

The training and testing errors of different ANN models are shown in table 4.1.

Table 4.1. The training and testing errors of ANN models

	NONE	exp	loc	con	exp+loc	exp+loc+con
size	0	1	4	20	5	25
training	2.804±0.0030	2.805±0.0066	2.830±0.0026	2.689±0.0016	2.745±0.0086	2.683±0.0072
testing	2.903±0.00052	2.805±0.0084	2.832±0.0026	2.695±0.0020	2.746±0.0078	2.689±0.0060

Each model is randomly initialized and trained 10 times.
Values are confidence intervals of relative entropy errors in *nats*
size: the size of the input layer
training: the cross-validation training error
testing: the testing error
NONE: the ANN model ignores any features in the structural environment description. The network output should be identical to the distribution of amino acids in the target.
exp: the ANN model employs the description of residue exposure.
loc: the ANN model employs only the description of local structure.
con: the ANN model employs the description of multiple contacts

Of the 109 structure-decoy pairs in 3 different sets, TUNE correctly detected 86 pairs (table 4.2). This overall performance is comparable to the pairwise distance dependent potentials of mean force with atomic structure description. Further, it is better than residue contact potentials RKBP (81/109) (Lu and Skolnick 2001) and CDF (75/109)

(Samudrala and Moult 1998).

Table 4.2. Evaluation of TUNE and other published potentials on decoy sets from ProStar

	size	PET2	RAM	RAM2	RAM3	KBP	RAPEF	RKBP	CDF	TUNE
ifu	44	8	30	37	28	32	30	22	21	31
misfold	24	16	24	16	23	24	24	24	19	24
asi	41	39	36	37	37	37	37	35	35	31
sum	109	62	90	89	88	93	91	81	75	86

size: size of decoy sets

decoy sets:
ifu: Independent Folding Units [v.1.0] by Moult and Unger (1991)
misfold: EMBL Deliberately Misfolded Set [v.1.0] Holm and Sander (1992), Mosimann et. al. (1995)
asi: Asilomar 94 Comparative Models [v.0.9], submitted by predictors to the CASP1 (Moult et. al. 1995)

potentials:
PET2: Physics-based Potential of Mean Force [v 2.0] by Avbelj (1992)
RAM: Pairwise Distance Dep. PMF - All Atoms, Residue Specific [v 1.0
by Head-Gordon and Brooks (1991)
RAM2: PMF - Brooks-Type Virtual Atoms, Non-Residue Specific [v 1.0]
RAM3: PMF - Brooks-Type Virtual Atoms, Residue Specific [v 1.0]
KBP: the atomic potential from Lu and Skolnick (2001)
RAPEF: the atomic potential from Samudrala and Moult (1998)
RKBP: the Residue contact potential from Lu and Skolnick (2001)
CDF: the residue-based potential from Samudrala and Moult (1998)
TUNE: our ANN model

For the seven decoy sets from Decoys'R'us, Figure 4.4 shows the correlation between the RMSD (root-mean-square-distance) and the score of compatibility for the decoy set of protein with PDB code 1ctf. The closer the decoy structure is to the native structure, its score is often higher. For this protein, the native structure has the highest score. Table 4.3 shows the performance of TUNE model compared with those of atomic energy functions developed by Gatchell and co-workers (2000) and by Lu and Skolnick (2001). For all 7

sets, TUNE performs better on the decoy set 2cro and 4rxn, worse on 1ctf, 1sn3 and 4pti, and obtains similar results on the other two sets. The overall performance is still comparable as measured by the values in table 4.2.

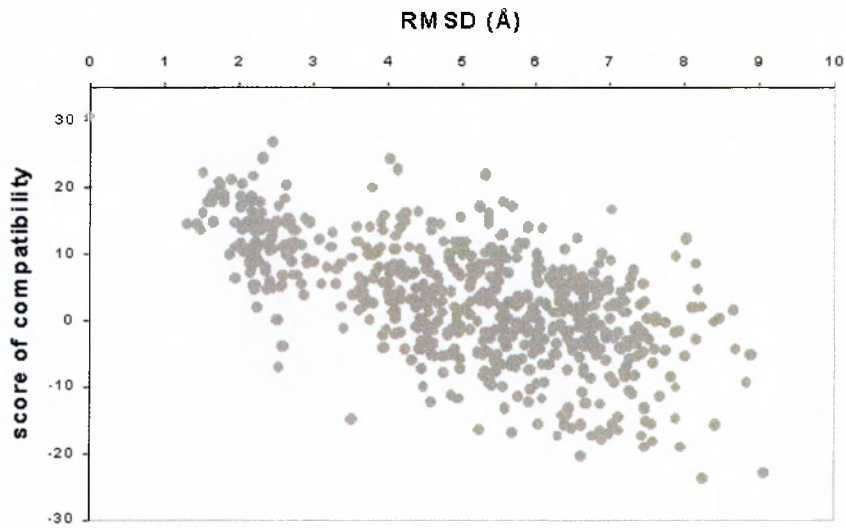


Figure 4.4. Correlation between the RMSD (root-mean-square distance) between the native and decoy structure after rigid-body superposition and the score of compatibility from my ANN model. The closer the decoy structure is to the native structure, its score is often higher. The dot at RMSD=0Å represents the native structure. The correlation coefficient is -0.61. The PDB code of this protein is 1ctf.

Table 4.3. Evaluation of TUNE, GDV and KBP on decoy sets from Decoy'R'Us

PDB code	1ctf	1ig9	1sn3	2cro	3icb	4pti	4rxn
TUNE	0.610	0.642	0.364	0.626	0.771	0.432	0.596
GDV	0.674	0.641	0.524	0.549	0.789	0.473	0.582
KBP	0.667	0.675	0.463	0.617	0.829	0.462	0.579

Values are absolute correlation coefficients between RMSD and potentials or scores of compatibility (see figure 4.3)
TUNE: our ANN model
GDV: the atomic potential developed by Gatchell and co-workers (2000)
KBP: the atomic potential developed by Lu and Skolnick (2001)

For these 7 decoy sets, my model does not always give the lowest score for the native model. Some decoy models can score lower. However, Vendruscolo and co-workers (2000) showed that for large enough databases, pairwise contact potentials could not stabilise all native folds equally well. However, such potentials are still useful for threading and other applications.

The rationale for noise removal and feature extraction was discussed in our previous work on the encoding of amino acid type (Lin et al. 2002a). Information theory has been employed in many different prediction problems (e.g. Schneider 1997). In figure 4.2, with analysis of the mutual information of neighbours, the co-operating effects of neighbours on the amino acid type of central residue are estimated. For simplicity, only

the two strongest signals in the contact description are considered. Unfortunately, the number of probabilities to be estimated increases exponentially with the number of considered amino acid types. I do not have enough data for finer encoding of the contact pattern. (A feature including 4 signals requires more than 3,200,000 probabilities to be estimated.) So, many features in the contact description are ignored: I assume that only the amino acid types of the closest neighbours affect the compatibility of the central residue. This is consistent with the optimal radius (4Å) being slightly larger than the radius of the largest side-chain i. e. Tryptophan (3.5 Å).

Figure 4.5 shows the effects of different combinations of features. The error of testing is related to the residue exposure. By adding the feature of local structure, the improvement on the accuracy of my ANN model is less dependent on residue exposure. When the central residue has no or a few neighbours, it is very accessible to solvent and hydrophobic amino acid types are often prohibited here. When the side-chain is surrounded by many neighbours, it is more likely to be a small amino acid like Glycine and Alanine. By including residue exposure, the performance of the predictor in such cases can be improved. However, on residues with intermediate numbers of neighbours, the predictor benefits less from this feature. Finally, I add multiple contacts. Compared with the ANN model using only local structure, this feature can assist the predictor on making better predictions especially when the residue is more buried.

I test several different schemes to describe secondary structure. For simplicity, I do not use the definitions from DSSP (Kabsch and Sander 1983), but utilise the information I

can get directly from PDB alpha carbon coordinates. The distances to the ±3, ±2, ±4 and their combinations are tested. Similar descriptions of local structure have been used for protein structure alignment programs (e.g. Taylor and Orengo 1989; May 1996). Regarding the cross-validation errors of ANN models, I conclude that any description of local structure including distances to ±4 alpha carbons could be sufficient and at least not much worse than any other combinations (data not shown). Owing to the flexibility of the ANN model, I can adopt an integrated continuous description for residue exposure, local structure and multiple contacts to neighbour atoms rather than discrete classes of structural environment employed in previous works (e.g. Bowie 1991, Overington et al. 1992).

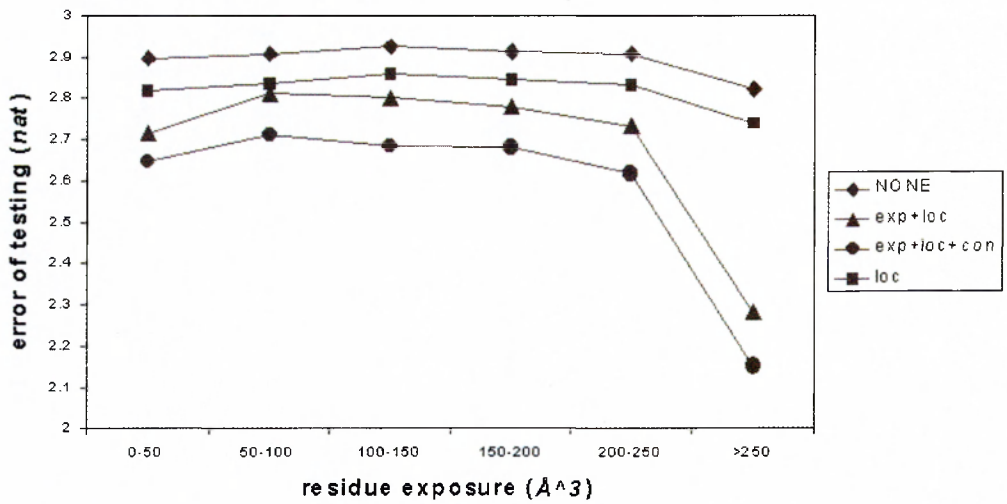


Figure 4.5. Performance of ANN models with different features.
The error of testing is the average cross-entropy error of ANN models on the testing set. The number of neighbours is the number of all neighbour atoms in the neighbour space.
NONE: the ANN model ignores any features in the structural environment description.
loc: the ANN model employs only the description of local structure.
exp+loc: the ANN model employs the description of residue exposure and local structure.
exp+loc+con: the ANN model uses the full structural environment description: residue exposure, local structure and multiple contacts.

In my structural description I use two spheres to describe each residue: the main-chain sphere and the side-chain sphere. Clearly, there are three types of contacts: side-chain-side-chain, side-chain-main-chain and main-chain-main-chain contacts. The ANN model is designed for use in threading (TUNE: Threading Using Neural Networks) where the template structures are native structures. Also, the conformation of their backbones will not be changed in the threading. Therefore, contacts between main-chain spheres will be preserved in the threading and so are ignored in the scoring scheme.

There are different methods for calculation of pseudo-energy. Knowledge-based mean-force pseudo-energy functions have focused on the probability of occurrence of contacts between atoms or residues (Sippl 1995). The basic assumption is that the conformation of protein structure follows the Boltzmann distribution: the probability of observing the contact is proportional to log energy states, and native protein structures should have lowest energy states. In pseudo-energy functions, calculation of probability of the reference states is a critical problem, closely related to the assumptions on protein structures (e.g. Skolnick et al. 1997). Another problem is that descriptions of atom-atom contacts ignore the orientation of their covalent bonds. These problems are avoided in my Bayesian approach. By using the occurrence of amino acids, which is much easier to obtain, I avoid calculation of probabilities in reference states. In my description of multiple contacts, most neighbours of the central residue are ignored. Yet the performance is still better than those of some residue contact potentials.

Scores from my approach are log-odds similar in form to widely-applied amino acid

substitution scoring matrices such as BLOSUM62 (Henikoff and Henikoff 1992). Compatibility of each aligned residue is explicitly described in my scoring function. These residue-specific log-odds can be conveniently employed for protein alignment algorithms such as the double dynamic programming algorithm (Taylor and Orengo 1989, Jones et al. 1992a) or other heuristic algorithms (e.g. Thiele et. al 1999).

Our model is not specially trained to discriminate native protein structures from decoy sets. However, the performance is still close to potentials with more detailed structural descriptions. We are testing it with the double dynamic programming algorithm in a full threading program (Taylor 1997a). In addition, I hope that this model can also be useful for other related applications such as comparative modelling, *ab initio* prediction and simulation of protein aggregation (Smith and Hall 2001).

In conclusion, I have described a means of assessing the compatibility of amino acid sequences for protein structure templates using an artificial neural network model. The performance of my model with a residue level structural description is comparable to those of pseudo-energy functions with atomic level structural descriptions and better than those of residue contact potentials.

Chapter 5.

Threading Using Neural Networks (TUNE): One-dimensional Profiles for Protein Structure-Sequence Alignment

To interpret biological functions of proteins from their amino acid sequences, a useful approach is to predict their three-dimensional (3D) structures first, since the functions of a protein is determined by the 3D co-ordinates of its atoms. To predict protein structures, fold recognition programs align probe protein sequences onto 3D structure templates. These programs were developed to detect remote evolutionary relationships, especially those undetectable by sequence alignment programs. Many fold recognition programs achieve these goals by including additional structural information from protein structure templates. A popular method is to generate Position-Specific Scoring Matrix (PSSM), sometimes called one-dimensional (1D) profiles, from protein structure templates.

I have shown that with an artificial neural network (ANN) model, I could generalise protein alignment samples and create amino acid substitution matrices (Lin et al. 2001).

In TUNE1D (Threading Using Neural Networks with one-dimensional profiles), the ANN model is extended to incorporate structural environment descriptions. Similarly, we can create PSSMs from protein 3D templates.

Our method is tested in CASP4 (the Fourth Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction). As one of the simplest methods, TUNE1D is a fast and useful template detector.

A TUNE1D server is available at http://mathbio.nimr.mrc.ac.uk/~kxlin/tune1d/

5.1 Introduction

While the increasing number of protein sequence data is being produced by the, now numerous, genome projects, biologists have huge databases of protein sequences, yet most of these sequences have no known biological function. Since the functions of proteins are decided by the 3D arrangement of their atoms, a classic approach to gain the knowledge of the biological functions of a protein is to solve its 3D structures via X-ray crystallography or NMR experiments. However, despite the improvements of structure determination techniques, these experiments are still expensive and time-consuming. Bioinformatic softwares can rapidly predict protein structures and functions using computers. Fold recognition programs align protein structures to sequences. When properly aligned to good structure template, sequences of unknown structures can reveal a large amount of useful information about the functionally and structurally important parts of proteins. To improve the detection of more remote homologies, methods have been developed to integrate structure and sequence features to generate alignments. Sequences can be aligned to known protein folds using energy functions or probabilistic scoring schemes (e.g. Jones 1992a, Bowie et al. 1991, Rice and Eisenberg 1997, Rost 1997, Jones 1999a, Kelley et al. 2000, Shi et al. 2001).

The protein 3D structures from X-ray crystallography and NMR experiments can be aligned and hierarchically classified (e.g. Murzin 1995, Orengo 1997). Experimental structures and their hierarchical classifications are often regarded as the standard of truth

for fold-recognition and other protein structure prediction programs (e.g. Brenner et al. 1998, Lackner et al., 1999).

Different methods have been developed to generate PSSMs (Henikoff and Henikoff 1994) using multiple alignments, (predicted) secondary structures and other features (e.g. Altschul 1997, Henikoff and Henikoff 1997, Elofsson et al. 1996, Rost et al. 1997, Zhang and Eisenberg 1994).

Bowie and co-workers (1991) calculated amino acid preferences for structural environments defined in terms of solvent accessibility, contact with polar protein atoms and secondary structure type. One-dimensional profiles, which can be aligned to sequences using a dynamic programming algorithm, were generated from protein structures based on these preferences.

In the 3D-1D substitution matrix approach of Rice and Eisenberg (1997), each structure position was defined by one of seven residue classes, three secondary structure classes and two burial classes; each sequence position was defined by one of seven residues classes and three predicted secondary structure classes. The matrix scores substitution between residues of different classes. A dynamic programming algorithm can use it to align a sequence probe with structures in a representative fold library after the prediction of probe secondary structures (Rost et al. 1997). In these programs, information from multiple sequence alignment of probe sequences is used to predict secondary structure and residue exposure. In recent successful approaches, multiple alignments of probe

sequences and target structures are used for the building of 1D profiles on both sequence and structure sides (e.g. Kelley et al. 2000, Shi et al. 2001).

Artificial neural networks have been applied in many bioinformatic problems (for a review, see Baldi and Brunak 1998). I have used an ANN model to generate amino acid substitution matrices (Lin et al. 2001). Owing to the flexibility of the model, a description of protein evolutionary distance was integrated into the construction of matrices, allowing generation of a series of matrices highly similar to the classic matrices like PAMs, (PETs) and BLOSUMs.

Here, I extend the model to include residue structural environment descriptions. I show that, with the additional structural information from protein 3D templates, prediction of residue substitution probabilities can be improved.

Another neural network is employed to assess alignment significance. I take an "integrated parameter optimisation" approach with this model. Parameter (gap penalties) optimisation and training of the alignment significance assessor are performed in one step. The program TUNE1D (Threading Using Neural Networks with one-dimensional profile) was tested in CASP4 (http://PredictionCenter.llnl.gov/casp4/Casp4.html) for the detecting of structure templates.

CASP (Moult et al. 1999, Moult et al. 2001) is a community-wide experiment to assess methods of protein structure prediction. Four previous experiments have been conducted

and reported in special issues of the journal PROTEINS: Structure, Function, and Genetics. As before, the goal is to obtain an in-depth and objective assessment of our current abilities in this area. In CASP4, 111 research groups and prediction servers submitted 5150 3D models for 43 target proteins, a total of 52 target domains were evaluated by the fold recognition assessor. As in previous CASPs, independent assessors evaluated the predictions (e.g. Lackner et al. 1999). Although TUNE1D is a very simple fold-recognition program, its performance in CASP4 is still satisfactory.

5.2 Methods

5.2.1 Data sets

The structure classification CATH (v2.0) (Orengo et al. 1997) was used to select training and testing sets for the ANN model. I consider only the first four CATH classes (mainly alpha, mainly beta, mixed alpha-beta, few secondary structures). Other classes, regarded as preliminary data in CATH, are ignored. 1938 pairs of domains are selected. 513 are pairs of domains from the same topology family but different homologous families. 1425 are pairs of domains from the same homologous family but different sequence families.

I align all 1938 pairs with SAP (Structure Alignment Program) (Taylor and Orengo 1989, Taylor 1999). In SAP, the pairwise similarity relationships between residues from different domains are scored on the spatial position of residues relative to the local co-ordinate frame. The score ranges from 0 to several hundreds and most significantly similar residue pairs score more than 1. Thus, to avoid noise from amino acids aligned without significant similarity, I set a threshold of SAP score to 1. Aligned residue pairs with lower scores are discarded (18% of all aligned residue pairs). Four fifths of these 1938 structure alignments (1538 alignments, 330531 aligned residue pairs) are randomly selected for training of the neural network, the remaining fifth (400 alignments, 82165 aligned residue pairs) for testing. Figure 5.1 shows the distribution of alignment sequence identity in the training and testing sets.

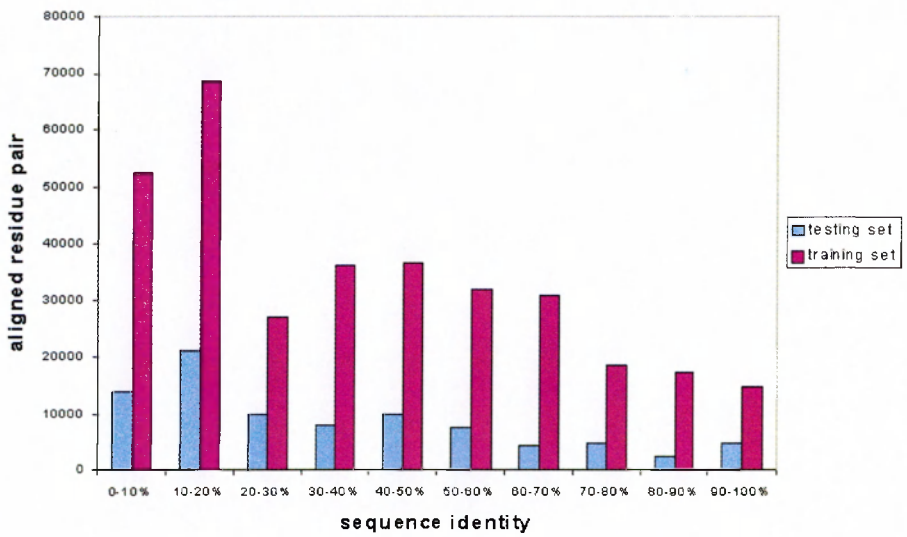


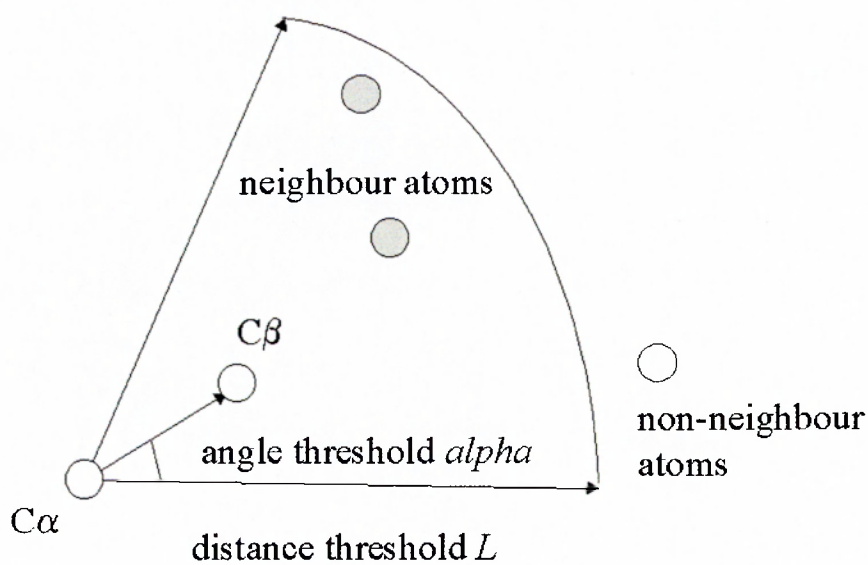
Figure 5.1. Distribution of testing and training sets over alignment sequence identity. Aligned residue pairs are randomly assigned to the testing and training sets. Sequence identities are from structure alignments.

5.2.2 Residue structural environment description

While in chapter 4 I introduced an encoding of residue structural environments, a more complex version is tested here to integrate more details of structural features into the description. In this description, contacts between residues are encoded using their orientations and distances, rather than volumes of overlapping regions. Each residue is described using two co-ordinates: the real alpha carbon atoms from PDB files and the

"centre" of the side-chain. I then divide space round the side-chain centre into 3 regions (fig 5.2.). Other side-chain centres or alpha carbon atoms in these regions are regarded as neighbours. To simplify this description, side-chains of Glycines are ignored. Further, neighbouring residue backbones are treated as side-chains of Asparagine, because of their similar chemical compositions. The angle threshold *alpha* is 60 degrees and the distance threshold *L* is 11Å (fig 5.2a.). I used the same process of parameter optimisation described in chapter 4.

a



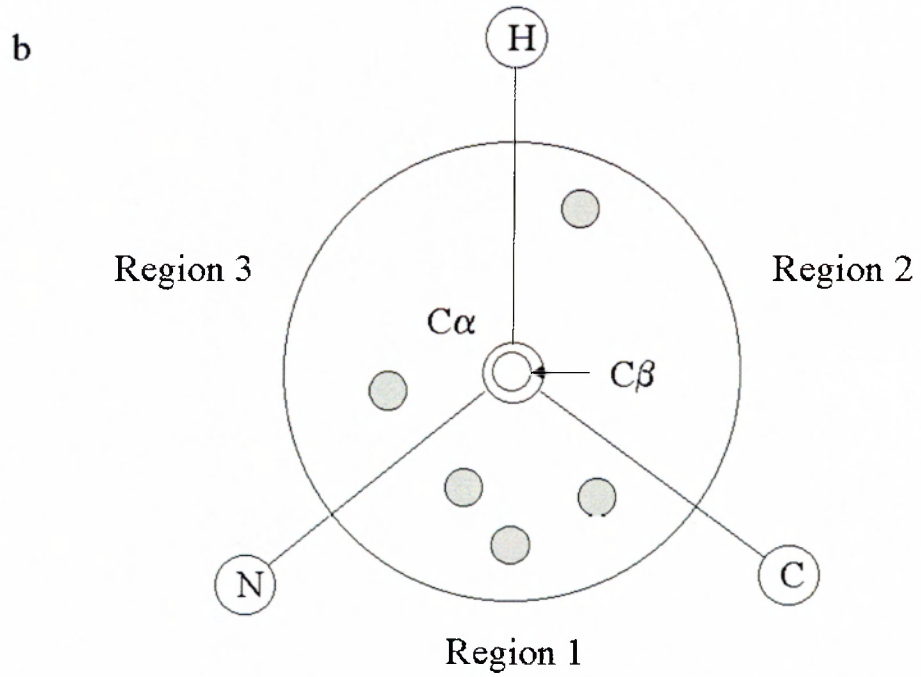


Figure 5.2. Encoding residue structural environment.
Each residue is described using two co-ordinates: the alpha carbon and the side-chain "centre". I then divide space around the side-chain centre into 3 regions. Other side-chain centres or alpha carbon atoms in these regions are regarded as neighbours. The angle threshold *alpha* is 60 degrees and the distance threshold L is 11Å. Residue structural environment is encoded by its local structure, residue exposure and contacts with other residues (see methods).

Given the angle and distance thresholds, I use 63 input units to describe the structural environment of a residue. One for the residue exposure, measured by the sum of all neighbours. The other two are distances from the alpha carbon to the alpha carbons of +4 and -4 residues on the sequence, describing local structure. 20 input units are employed to

encode neighbours in each region. For each neighbour, a value is added to the corresponding unit according to its amino acid type. The distance threshold *L*, minus the distance from the side-chain centre to the neighbour, weights the value, so closer neighbours have greater influence.

I used only the first three units, including residue exposure and local structure for the simplified version of TUNE1D in CASP4,

5.2.3 Generation of 1D profiles

For each aligned residue pair in my training set, a residue is characterised by 84 input units: 1 for the sequence identity of the alignment, 20 for the amino acid type of this residue, and the other 63 units for its environment description. Given the large mount of data, an orthogonal encoding was used (fig 3.2) for better performance. Lower dimensional encoding of the amino acid types was not found to be beneficial.

An ANN model with 84 input units, 30 hidden units and 20 output units was trained to minimise the average difference between the predicted substitution probabilities and the amino acid type of the residue, which is aligned to the first residue (fig. 5.3). The training algorithm and parameters (training rate and momentum) were described in chapter 2 (Lin et al. 2001).

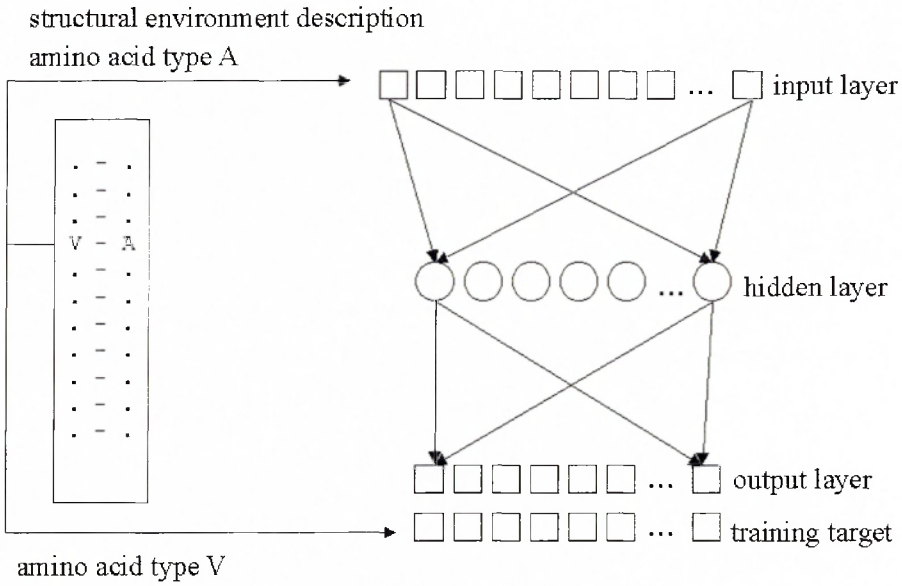


Figure 5.3. Training and testing of the neural network.
A pair of aligned amino acids was used here. I enter the Valine and its structural environment description into the input layer. The training target is set to the Alanine, encoded by the orthogonal encoding scheme. The back-propagation training algorithm modifies the connection weights to minimise the difference between the output layer and the training target.

I have 2667 CATH sequence family representative domains in my template database. For each domain, the structural environment, amino acid type of each residue, and a default sequence identity is described and presented to the trained ANN model. I predict the substitution probabilities of residues via propagation of the model. The log-odds PSSMs (Position Specific Scoring Matrix) can then be calculated as:

$$Sxy = \ln (P(y|x)/Ry) \quad (1)$$

Sxy, the score of residue x aligned to amino acid y is a function of the probability P(y|x),

which is the probability of x (with its structure environment) substituted by amino acid y, and Ry, the occurrence of amino acid type y.

I set sequence identity to 15% for generation of default profiles. The rationale for this number is explained in the discussion section.

5.2.4 Parameter optimisation and alignment significance accessing

A probe sequence is aligned to 1D profiles using a global-local dynamic programming algorithm, a global sequence alignment algorithm without any penalty for end-gaps (Taylor 1988, Fischer and Eisenberg 1996).

I take the approach of Jones (1999a) to evaluate alignments. Another ANN model is trained for this task. Given a pairwise alignment of two domains in my CATH representative set, I describe the significance of this alignment as the P, which is given by

$$P=S_{sub}/S_{all} \quad (2)$$

where S_{sub} stands for size of the smallest sub-class that includes the two domains, and S_{all} is size of the whole non-redundant set (here it is 2667 for CATH sequence families). P is the probability of finding another domain, which shares a same or higher level of similarity with one of the domains by random selection in my non-redundant structure set. Obviously, the range of P is (0,1). If two domains are the same, the P value of the self-alignment is $1/S_{all}$ (1/2667). If two domains are from two different main classes, the P value is S_{all}/S_{all} =1.

For any pairwise alignment from my representative set, I can have the predefined P value according to the hierarchical classification of CATH. I use this value as the target of the neural network training. The ANN model is trained to predict P with alignment length, profile length, sequence length, and alignment score from TUNE1D. This neural network has 4 input units, 4 hidden units and 1 output unit. For the training of this neural network, I randomly select 8034 pairs of domains, and used a 6-fold cross-validation approach. I optimise the alignment parameters (gap penalties) of TUNE1D to minimise the average difference between the ANN output and the P value.

5.2.5 TUNE1D in CASP4

In CASP4, 3D models of probe sequences were generated using an integrated approach. Our group used QUEST (Taylor 1998) and TUNE1D to select target structure templates. Results were manually checked with the scores of significance. A structure template could be taken if the alignment seems reasonably good, there are functional relationships between the probe and the template or the score of significance from TUNE1D is less than 0.20. Normally, only the top hit by TUNE1D will be used. With predicted secondary structures (Frishman and Argos 1997) from the multiple sequence alignments (Taylor 1998), the backbone alpha carbon structures were then built using the Multiple Sequence Threading program MST (Taylor 1997a). I then obtained full atom structure models from the MaxSprout server (Holm and Sander 1991).

5.3 Result and Discussion

5.3.1 1D profiles

There are many different approaches to generate 1D profiles or PSSMs (Henikoff and Henikoff 1994). It is assumed that by adding structural information or information from a multiple alignment, more accurate predictions of residue substitutions can be made. By making more accurate predictions, one can generate better quality alignments and detect more remote relationships between proteins than with sequence alignment alone.

Different scoring matrices have been constructed for making protein sequence alignments. Dayhoff and co-workers produced the classic PAM (Point Accepted Mutation) matrices from a Markov model of amino acid substitution (Dayhoff 1978). In their model, evolutionary distance was scaled with the unit PAM. Matrices with higher PAM values, like PAM250, are better suited for scoring amino acid substitutions in alignments of lower sequence identities or longer evolutionary distances, while PAM80 is more appropriate for aligning very similar sequences. Similar results are obtained with comparable matrices. To detect remote relationships between proteins, it is often suggested to use matrices generated from alignments of more remotely related proteins (e.g. BLOSUM40).

In our previous work on the SMN (Substitution Matrices from Neural networks) (Chapter

2, Lin et al. 2001), a method to generate amino acid substitution matrices from protein structure alignments was described. By using an ANN, it is possible to generalise the evolutionary relationships between amino acids and generate amino acid substitution matrices at any given evolutionary distance.

This flexible ANN model has been extended to include the residue structural environment into prediction. Figure 5.4 shows the increase in prediction accuracy by including different structural environment descriptions. By introducing descriptions of residue exposure, local structure and multiple contacts, the error of the prediction is reduced.

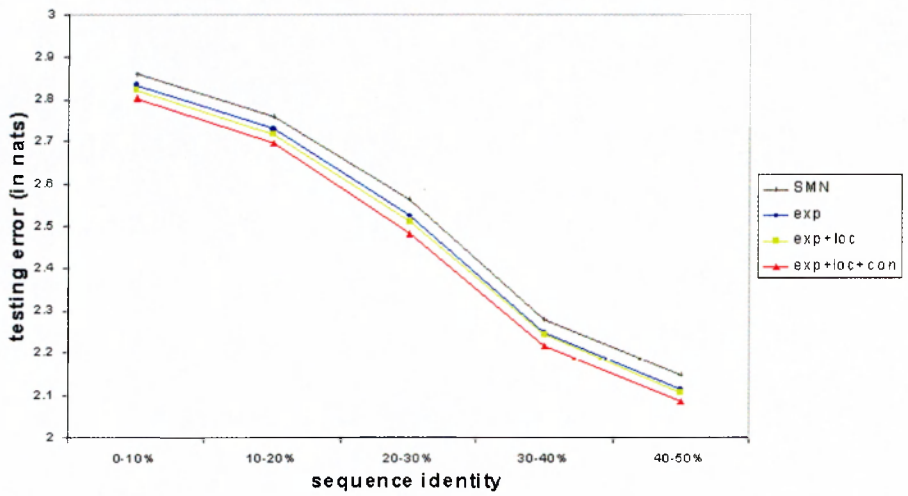


Figure 5.4. Performance of the ANN models with different structural features over alignment sequence identity.
The error of testing is the average cross-entropy error of ANN models on the testing sets in *nats*.
SMN: the ANN model employs only the amino acid type.
exp: the ANN model employs the description of residue exposure.
loc: the ANN model employs the description of local structure.
con: the ANN model employs the description of multiple contacts.

Using 3D-1D matching methods, fold recognition programs aim to detect remote relationships even when sequence alignment programs fail to. Most scoring matrices for this purpose are derived from alignment examples of distantly related proteins. Sequence identities of most alignment examples are lower than 30% (e g. Rice and Eisenberg 1997, Blake and Cohen 2001). Another approach is to weight the training set according to alignment sequence identity so that the matrix is biased to lower sequence identity (e.g.

Shi et al. 2001). With the ANN model, in our training set, I can use alignment examples of very different sequence identities. However, to generate 1D profiles, the default sequence identity is still an important parameter. Figure 5.5 shows the influence of the default sequence identity on prediction accuracy on different sets. By setting higher sequence identity, the ANN works better on alignments of more similar sequences. Nevertheless, I assume the range of sequence identities of TUNE1D alignments should lie mainly between 10-20%. To optimise the performance, the default sequence identity is set to 15%.

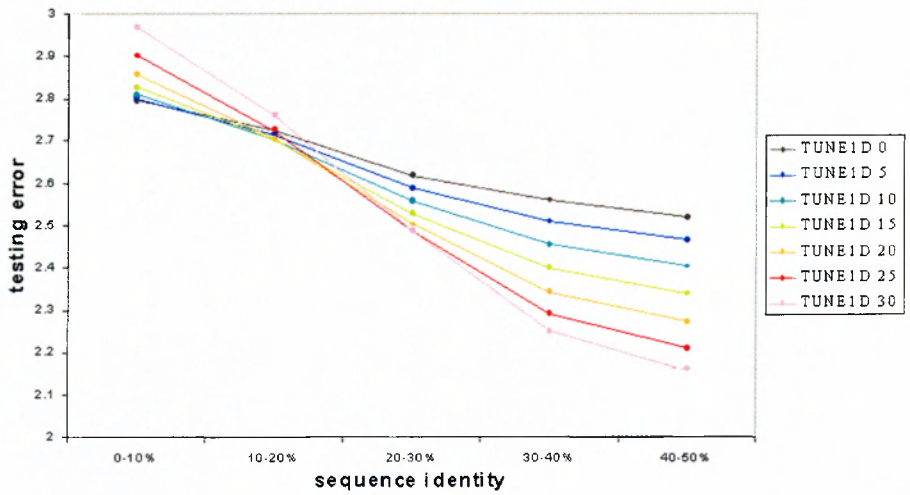


Figure 5.5. Performance of the TUNE1D model with different sequence identity settings.
The error of testing is the average cross-entropy error of the TUNE1D model on the testing sets in *nats*. The numbers in the legend is the default sequence identity of the TUNE1D model for the generation of 1D profiles.

The results are shown in *nats* of relative entropy in figure 5.4 and figure 5.5. I obtained similar results with RMS (Root-Mean-Squared) error function.

In the 3D-PSSM approach of Kelly and co-workers (2000), 3D profiles from structure templates are generated using multiple structure alignments of superfamily members. Similar profiles can be generated from HMM and multiple sequence alignments (for a

review, see Durbin et al. 1998). In theory, this approach could generate more accurate profiles because of consideration of family-specific conservation and substitution patterns. However, it is limited by the quality of multiple alignments, which are related to the distribution of family members (May 2001). The TUNE1D model generalises substitutions of amino acids regarding their structural environments. It requires no structure classification or structural alignments, and a profile can be generated from any single protein structure. However, to improve the performance, I am developing protocols to modify TUNE1D profiles with knowledge from structure and sequence alignments.

5.3.2 Assessing alignment significance

The classical approach to assess alignment significance assumes that gapped alignments scores of random sequences follow the extreme value distribution (Altschul and Gish 1996). Jones (1999a) introduced another approach to assign alignment significance. He randomly selected pairs of protein domains with known 3D structures. If the two domains of a pair are from the same topology family in CATH, the hierarchical classification of protein structures, the target value of alignment significance is set to 1, otherwise to 0. For each pair, he collected the lengths of two domains, the alignment length, the alignment score, and scores of sequence-structure compatibility from pseudo energy functions. For the fold recognition program GenTHREADER, he used these scores to predict the significance of the alignment. An ANN model was trained for this task. The neural network training algorithm changes the connection weights of the model to minimise the average difference between the target score and the ANN output value. In

this approach, the significance of alignment is known before the generation of alignment, and the ANN model tries to predict it with various scores from the fold recognition program. Because of the flexibility of the ANN model, alignment significance can be assessed with very different entries. And the significance of each alignment can be assessed with only one value. A similar approach has been used to identify homology in protein structure classification (Dietmann and Holm, 2001).

This approach has been extended by using the P value as training target, instead of the binary value used by Jones. It is hoped that this will give a more detailed description of relationships between protein domains.

With this approach, the overall performance of our program can be judged via only one value: the cross-validation error of the assessing ANN model. It specifies how much the output of the fold-recognition program confirms the hierarchical structure classification (table 5.1).

Table 5.1 cross-validation errors of the neural network on the assessing of alignment significance

ext\open	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6
0.0	0.03895	3877	3873	3847	3833	3854	3859	3850	3840
0.1	3898	3876	3894	3845	3823	3844	3843	3846	3872
0.2	3903	3861	3852	3868	3852	3882	3890	3889	3886
0.3	3898	3887	3901	3891	3834	3881	3878	3890	3895
0.4	3900	3888	3879	3914	3879	3923	3880	3928	3930

ext: gap extension penality
open: gap open penality

Because the overall performance of a fold-recognition program can be described with a single value, we can easily optimise program parameters to decrease this error. In this case, the parameters are gap penalties. Nevertheless, other parameters can also be optimised in the same framework.

Here the optimisation of TUNE1D totally ignored the quality of alignments. That is one of the reasons why I did not use alignments from TUNE1D in CASP4 for the generation of 3D models. However, some alignments generated by TUNE1D show reasonable quality under manual examination.

TUNE1D makes 2667 pairwise alignments between 1D profiles and a middle-size protein

probe sequence (150 amino acids) in 12 seconds on a PC with a Pentium III 800 chip. (The compiler is GNU g++ 2.95.2. The operating system is Linux 2.4.) It is a very fast program because of its simplicity: It is a pairwise profile-sequence alignment program with an ANN significance assessor. TUNE1D does not build multiple alignments on the sequence or structure side. No predictions of secondary or solvent exposure are included. No correlated mutations of residues are considered. Also it does not construct hidden Markov models or templates from iterative searching. In CASP4, I did not even include residue contacts in the structural description. However, by combining TUNE1D and MST (Taylor 1997a), we hope we can integrate multiple alignment and consideration of correlated mutations to improve performance.

5.3.3 TUNE1D in CASP4

In CASP4, TUNE1D used only exposure and secondary structures to help prediction of substitution probabilities because the encoding of residue structural environments was still under investigation, and a simpler version is faster. Compared to other fold recognition programs, TUNE1D is faster because the amount of computation is extremely small.

Nevertheless, in CASP4, the performance of TUNE1D, when used in concert with other programs (see chapter 7), is better than many much more complex programs. For 9 structures generated using templates found by TUNE1D, 3 of them are ranked about average, the remaining 6 are in top quarter of all submitted models. Also, TUNE1D

detected good target structures for homologous, analogous and new fold probes (table 5.2). I think the highly automatic procedures of 1D profile generation and parameter optimisation are key for improving accuracy.

Table 5.2. TUNE1D in CASP4

	code	rank	num	eqr1	rms1	length
FR/H	100	28	167	175	2.4	250
	109	103	235	46	3.4	182
	110	187	293	34	2.9	128
	121_2	30	202	41	3	126
FR/A	108	31	212	83	2.8	203
	114	93	219	36	3.2	87
	126	32	258	41	3.1	163
FR/NF	087_2	11	177	47	3.4	120
	105	61	244	31	3	94

FR/H: Fold Recognition/ Homologous
FR/A: Fold Recognition/ Anologous
FR/NF: Fold Recognition/ New Fold

code: target code
Best match between target and template (Fold correctness) derived from ProSup (Lackner et al. 2000) superimpositions:
rank: rank by the number of structurally equivalent residues
num: number of submitted models
eqr1 : number of structurally equivalent residues
rms1 : corresponding rmsd

Details of some of these targets can be found in chapter six, while a further "in house" application is described in chapter 7.

Chapter 6.

TUNE1D in CASP4: the critical assessment of protein structure prediction

This chapter describes some results of blind predictions submitted to the fourth round of Critical Assessment of Structure Prediction (CASP4). Models were constructed via automatic methods and manual examinations. TUNE1D, together with other programs, were tested in our procedure. As expected, the behaviour of TUNE1D is very similar to some 3D-1D matching fold recognition programs. Also, it is demonstrated that TUNE1D is a useful tool for template detection. The results suggest that among others, careful manual modifications of alignments, good structure refinement and loop construction methods are needed for better predictions.

6.1 Introduction

Every two years since 1994 sequences of some unpublished protein structures are collected from X-ray crystallographers and NMR spectroscopists. These sequences (prediction targets) are made available on a web site by CASP organisers. Groups of predictors build structural models of these proteins without knowing their experimental structures. Because these structures are not available before the deadline of model submission, it is impossible to include them into data sets of prediction programs and participants can only make "blind" predictions. Also, each model is evaluated via many quality measures (e.g. Zemla et al. 2001, Cristobal et al. 2001). The CASP experiment (Moult et al. 2001) is expected to evaluate objectively protein structure prediction methods. As I stated before, it is one of the most important experiments in bioinformatics.

Currently, many bioinformatic tools are available online. Some of them were tested in CAFASPs (Critical Assessment of Fully Automated Structure Prediction experiment) and produced good results (Fischer et al. 2001). However, human intervention is allowed in CASP for making predictions. Manual examination and modification of alignments and evaluation of model quality are still crucial for accurate predictions (e.g. Bates 2001, Murzin and Bateman 2001). We used an integrated approach in CASP4. Results from fully automatic servers and programs were manually inspected before construction and submission of models.

6.2 Methods

6.2.1 Sequence databank searching

The iterative sequence databank search tool QUEST, originally written by Taylor (1998), was used for finding candidate homologues. QUEST results are built up over a series of profile-based searches guided by multiple alignment. It is expected to detect distantly related sequence hits using sequence information only.

6.2.2 Secondary structure prediction

Predictions from servers including PSIPRED (Jones 1999b), PHD (Rost and Sander 1993) and SSpro (Baldi et al. 1999, Baldi et al. 2000b) were collected and manually examined. All three programs predict secondary structures from sequence multiple alignments using artificial neural network models. PSIPRED and PHD use the classic approach of employing local information. SSpro uses bi-directional recurrent neural networks to include long-range interactions.

6.2.3 Detection of structure templates

The programs FFAS (Ychlewski et al. 2000), GenTHREADER (Jones 1999a), 3D-PSSM (Kelley et al. 2000), 123D (Alexandrov et al. 1996), TUNE1D (described in chapter 4) and others were used to detect structure templates. The CATH (v1.7) assignments and

function annotations of the detected templates were collected and compared for selection of templates. Because one of our aims is to test TUNE1D, the decision was often biased to TUNE1D results.

6.2.4 Multiple alignment and construction of alpha-carbon models

The multiple alignment program MULTAL (Taylor, 1988) and the threading program MST (Taylor, 1997) were used to generate multiple alignments and alpha-carbon models. Although some alignments are examined by eye, we did not manually modify them.

6.2.5 Construction of full atom models

After selection of alpha-carbon models by hand, we used MaxSprout (Holm and Sander 1991) web server to construct full atom models.

6.3 Results and discussion

6.3.1 Target T100

TUNE1D easily recognised the template for this target. The model was based on the structure of pectate lyase C from *erwinia chrysanthemi* (PDB code: 1air). Both the target protein and the template have the same right-handed beta-helix fold. Although the loops outside the helix were not accurately placed, our model for this target has a reasonable score. T100 is considered an easy target. Despite this, the HMM method of Karplus et al. (2001) only managing to achieve moderate success after careful human intervention. 3D-1D matching fold recognition programs like 3D-PSSM (Bates et al. 2001), FUGUE (Williams et al. 2001) and TUNE1D all confidently recognised good templates.

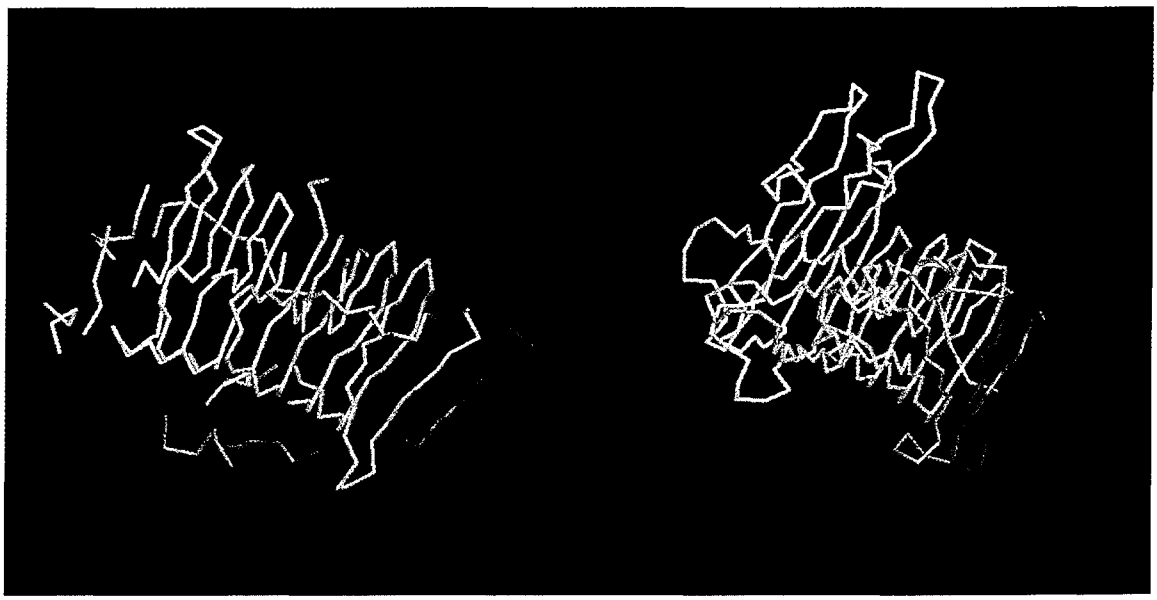


Figure 6.1 Threading of target T100
It shows our model (left) and the experimental structure (right) of the target T100. Both have a right-handed beta-helix fold. The N terminals were coloured in blue while warmer colours were used for following residues.

6.3.2 Target T108

Target T108 adopts a jelly roll beta-sandwich structure. Our model was based on the N-terminal Cellulose-binding domain from *cellulomonas fimi* (PDB code: 1ulo), which was correctly found by TUNE1D. However, the template (152 residues) is shorter than the target sequence (206 residues). The N-terminal of the sequence was modelled as random coil and placed at the side of the domain. As expected, this part of model is not similar to the corresponding part of the experimental structure. However, the topology of the structure was correctly predicted and most strands can be aligned between the model and the structure. Loops in our model show little resemblance to the native loops. With the same template (1ulo), Williams et al. (2001) built a better model with major manual modifications of alignments. Gap positions are optimised according to secondary

structure predictions. Their model also benefits from a more careful procedure of loop modelling (Rufino et al. 1997).

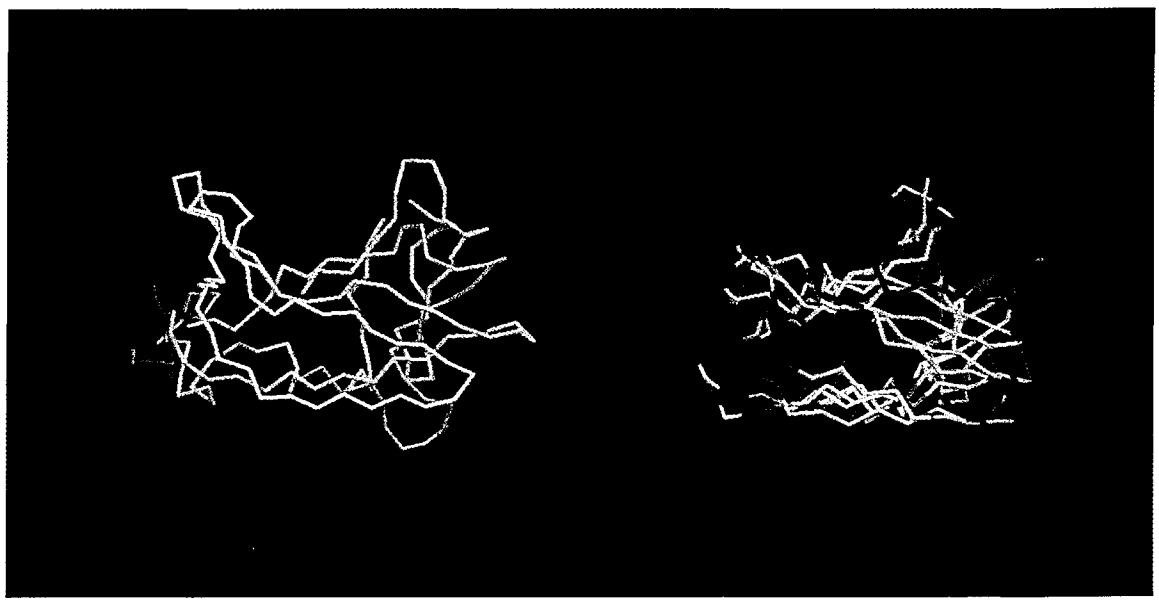


Figure 6.2 Threading of target T108
It shows the model (left) and the superposition (right) of the model and the target structure. The structural alignment is coloured according to the structural similarities between residues. Warmer colours are used for more similar regions. Blue is used for parts that are not aligned. The most similar regions are coloured in red.

6.3.3 Target T114

Target T114 has a gamma-crystallin like fold. With no classic crystallin sequence/structure motifs, it was a hard target for prediction. The template found by TUNE1D was the translational elongation factor G from *thermus thermophilus* (PDB code: 2efg). The overall folds are globally similar. Both are mainly beta proteins. However, the lengths and relative orientations of strands are significantly different. 2efg has a beta barrel fold while the target can be classified as a beta sandwich. Secondary

structure predictions correctly implied that target is a mainly beta protein. Because the template and the target have very similar composition and 1D organisation of secondary structures, it is very easy for 3D-1D matching programs to give misleading results. The group of Galaktionov, Nikiforovich and Marshall built the best model for this target. They used an *ab initio* procedure with predicted residue-residue contact matrices. Many 3D-1D matching or HMM approaches failed to produce accurate models for this target. Hopefully, good threading programs should be able to achieve good results on such cases with consideration of residue contacts.

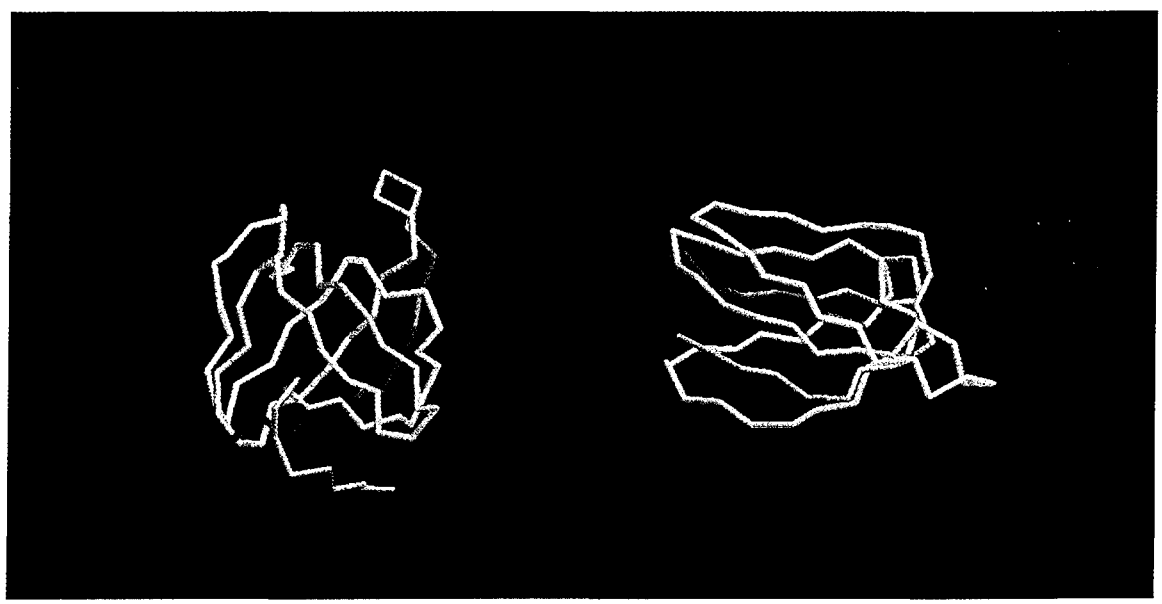


Figure 6.3 The template for target T114
The template (left, PDB code: 2efg) detected by TUNE1D for the target T114 (right). The template is a beta-barrel structure while the target has a beta sandwich fold. The N terminals were coloured in blue and warmer colours were used for following residues.

6.4 Conclusion

Three examples of prediction results were given in this chapter. The first target is a classified as a fold recognition homologous target. The other two are fold recognition analogous targets. While our procedure produced a good model for the easy target, predictions of the second and the third targets are less successful.

TUNE1D is perhaps the simplest and fastest program in CASP4. The program has only about 1000 lines of code. It makes pairwise alignments between the probe sequence and the PSSMs from a template library. And in the template library, PSSMs are generated using only residue exposure and local secondary structure. It takes only about 10 seconds for one sequence. However, it is interesting to see that a program without using multiple alignments, iterative database searching or hidden Markov models can still detect good templates sometimes. I think the automatic procedure of parameter optimisation described in chapter 4 is one of the reasons. Nevertheless, programs using multiple alignments on the template and probe sides like 3D-PSSM performed significantly better. And human intervention is clearly improving the results, especially for fold recognition homologous targets (e.g. Bates 2001, Williams et al. 2001). For more difficult targets like T114, we expect threading programs to perform better.

Chapter 7.

NusA: a case study

In *Mycobacterium tuberculosis* NusA (N Utilization Substance protein A) is an RNA-binding transcriptional regulatory protein within the RNA polymerase complex. It modulates several of ribosomal RNA transcription processes, influences both the rate of RNA chain elongation and the efficiency of termination.

The X-ray crystal structure of NusA was solved by Gopal et al. (2001) in the Division of Protein Structure and the Division of Mycobacterial Research of the National Institute for Medical Research. Before publication of the structure, we were given its sequence. Without knowing the experimental structure, we tested TUNE1D and other prediction programs in this "mini CASP" experiment.

7.1 Introduction

After the initial synthesis of RNA at a promoter, RNA polymerase (RNAP) regulates the elongation and termination of RNA chains, and becomes responsive to N utilization substances. The transcription factor NusA has been shown to interact physically with RNAP (Greenblatt and Li, 1981) and RNA (Mah et al. 2000). It pauses transcribing RNAP and participates in termination and antitermination of transcription. Gibson and co-workers (1993) found S1 homology region and duplicated KH homology regions in NusA by sequence comparisons. S1 and KH domains are found in proteins that associated with RNA specifically and nonespecifically. While there was strong evidence to suggest that these domains interact with RNA, the C domain of this protein was also found to bind to the α subunit of RNAP, as well as the N gene antiterminator protein (Mah et al. 2000). Together with these proteins, NusA plays an important role in the elongation, termination and antitermination of RNA synthesis (Burns et al. 1998, Zhou et al. 2001, for a review, see Weisberg and Gottesman 1999)

7.2 Methods

A variety of methods were used to learn more about the probe protein sequence.

7.2.1 Sequence information

Protein: NusA (from *Mycobacterium tuberculosis*)

Length: 347 residues (the flexible linker between the N-terminal domain and S1 domain, and the end of the C-terminal domain are not covered in the solved structure, see figure 7.1)

Sequence:

```
MNIDMAALHA IEVDRGISVN ELLETIKSAL LTAYRHTQGH QTDARIEIDR KTGVVRVIAR
ETDEAGNLIS EWDDTPEGFG RIAATTARQV MLQRFRDAEN ERTYGEFSTR EGEIVAGVIQ
RDSRANARGL VVVRIGTETK ASEGVIPAAE QVPGESYEHG NRLRCYVVGV TRGAREPLIT
LSRTHPNLVR KLFSLEVPEI ADGSVEIVAV AREAGHRSKI AVRSNVAGLN AKGACIGPMG
QRVRNVMSEL SGEKIDIIDY DDDPARFVAN ALSPAKVVSV SVIDQTARAA RVVVPDFQLS
LAIGKEGQNA RLAARLTGWR IDIRGDAPPP PPGQPEPGVS RGMAHDR
```

7.2.2 Sequence databank searching

The databank searching scheme QUEST was originally developed by Taylor (1998). It was then fully commented, re-written and released by Kleinjung, Hatwell and Brown (Taylor and Brown 1999, Kleinjung et al. submitted). QUEST is an iterative sequence databank search tool guided by multiple alignment. It tries to detect distantly related

sequences with a fast heuristic searching algorithm.

7.2.3 Secondary structure prediction

PSIPRED (Jones 1999b) is a secondary structure predictor using PSSMs (Position Specific Scoring Matrices) from sequence databank searching program PSI-BLAST (Altschul et al. 1997). We used the web server of PSIPRED (v. 2.0) at http://insulin.brunel.ac.uk/psipred/.

7.2.4 Recognition of repeats

We used the REPRO web server (George and Heringa, 2000) at http://mathbio.nimr.mrc.ac.uk/~rgeorge/repro/ to detect repeats in the sequence. However, the repeats of KH domains were manually recognised.

7.2.5 Detecting structure templates

Different "cuttings" of domains were tested on the sequence. For each domain candidate (sequence segment), we searched the template library of TUNE1D, which includes 2667 domains from CATH (Orengo et al. 1997) (v2.0) for domains with similar structures. This process continued until we found confident hits for most segments. The web server of TUNE1D is available at http://mathbio.nimr.mrc.ac.uk/~kxlin/tune1d/.

7.2.6 Multiple alignment and construction of alpha-carbon models

The multiple alignment program MULTAL (Taylor, 1990) and the threading program MST (Taylor, 1997) were used to generate multiple alignments and alpha-carbon models for each domain.

7.2.7 Quaternary structure

After construction of models for each domain, we manually combined the models of all four domains and the complex was docked to a segment of RNA 3D structure by hand.

7.3 Results

Most secondary structures were correctly predicted. Two strands (residue 141-145, 319-327) and one helix (122-126) were missed, a small helix (231-236) was wrongly predicted as strand. The prediction confidence given by the program in these regions is often lower than average. The three-states accuracy of this prediction is 80%.

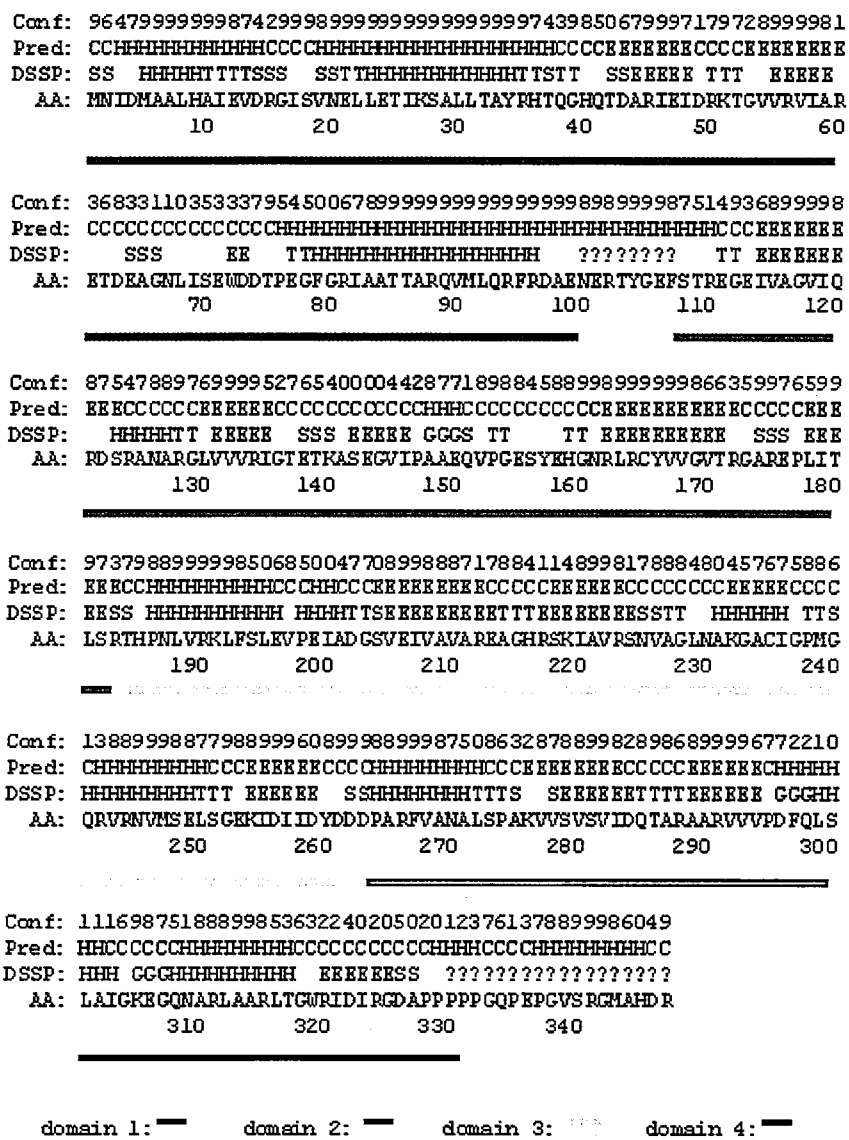


Figure 7.1. PsiPred prediction results
Conf: Confidence (0=low, 9=high)
Pred: Predicted secondary structure (H=helix, E=strand, C=coil)
DSSP: secondary structure defined by DSSP ('?' = residues not covered in the experimental structure).
AA: Target sequence

Neither the sequence databank searching by QUEST over PDB50 sequence databank (Taylor 1998) or the detection of repeats by REPRO gave confident results.

From the prediction of secondary structure and the results of fold recognition of different segments, four domains were manually assigned: the N terminal domain (residue 1-100), the S1 domain (109-183) and the two KH domains at the C terminus (184-262, 263-329).

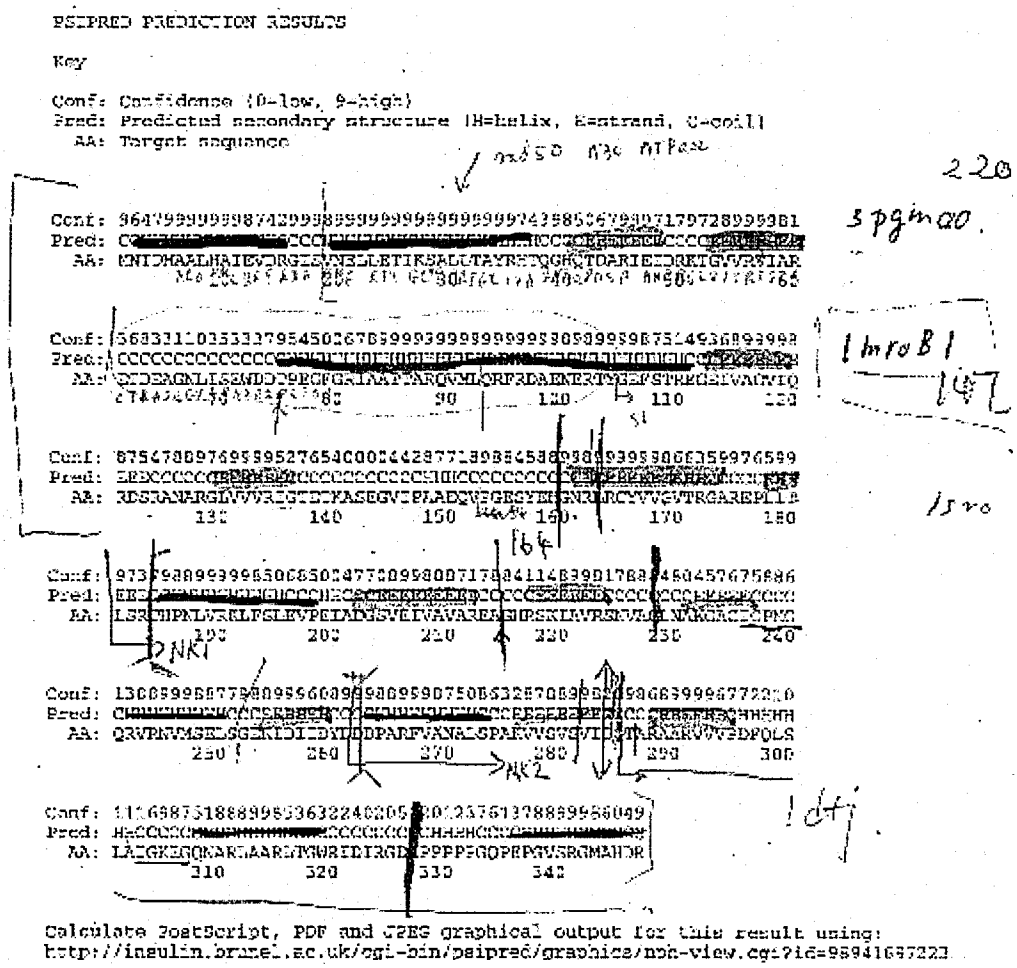


Figure 7.2. A working draft of the domain assignment
Domain boundaries are predicted according to the results of secondary structure prediction and fold recognition.

For each sequence segment, the structure template detected by TUNE1D and similar sequences detected by QUEST in NR (non-redundant sequence database) were aligned with the program MST using the double dynamic programming algorithm. It also built the alpha-carbon models of the sequence segments. For the N terminal domain, the model is dissimilar to the real structure (RMSD 16.2Å over 78 residues). For the other three domains, satisfactory models were generated (RMSDs are 2.4 Å over 68 residues, 2.2 Å over 69 residues and 2.2 Å over 64 residues). The lengths of the four domains are respectively 100, 74, 77 and 67 residues.

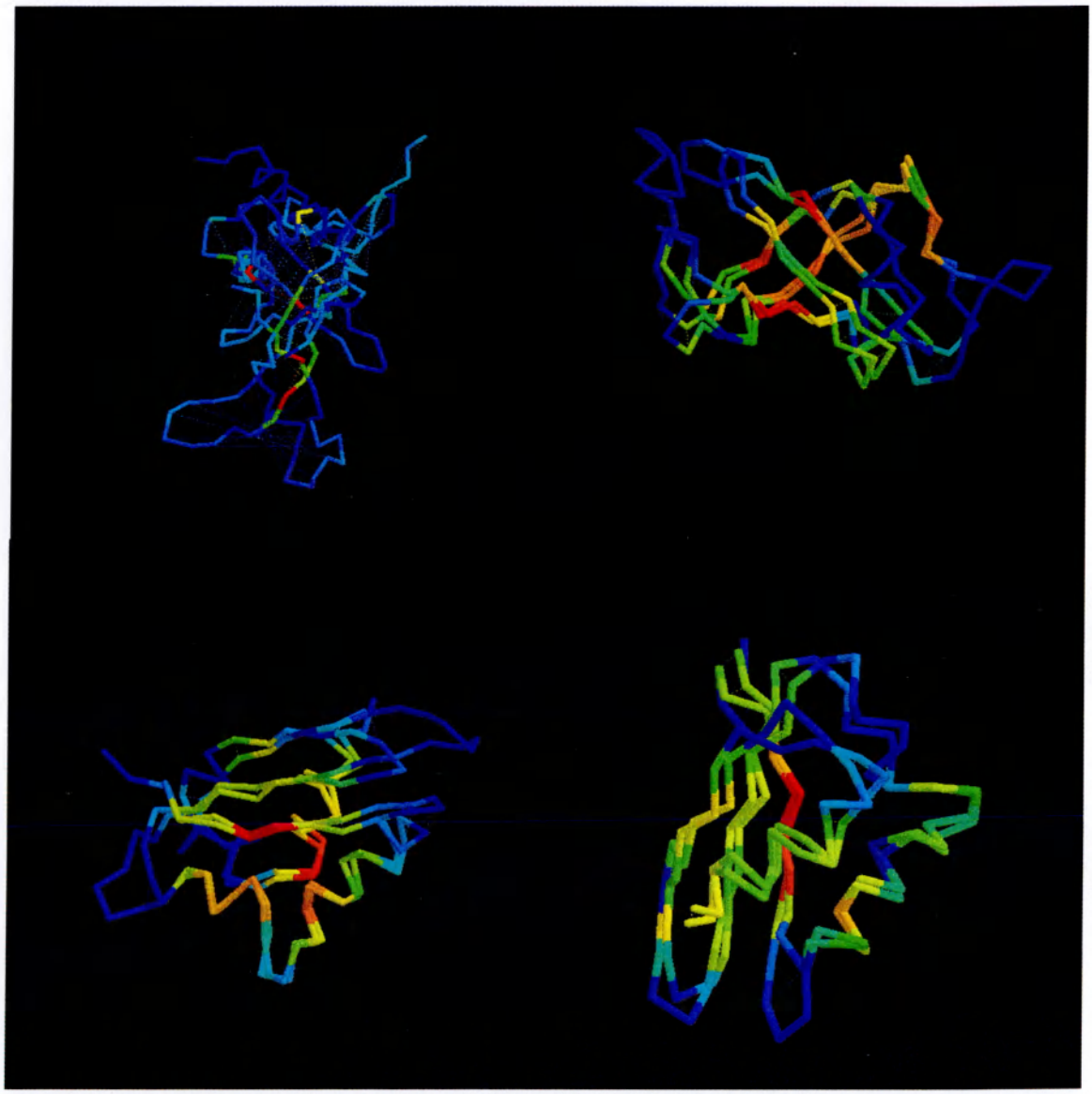


Figure 7.3. SAP structure alignments between the models and the experimental structure.
From N-terminal to C-terminal: domain 1 (top left), domain 2 (top right), domain 3 (bottom left) and domain 4 (bottom right).

Not surprisingly, the quaternary structure constructed from the manual combination of domains is of poor quality. Relative orientations of domains are wrongly predicted.

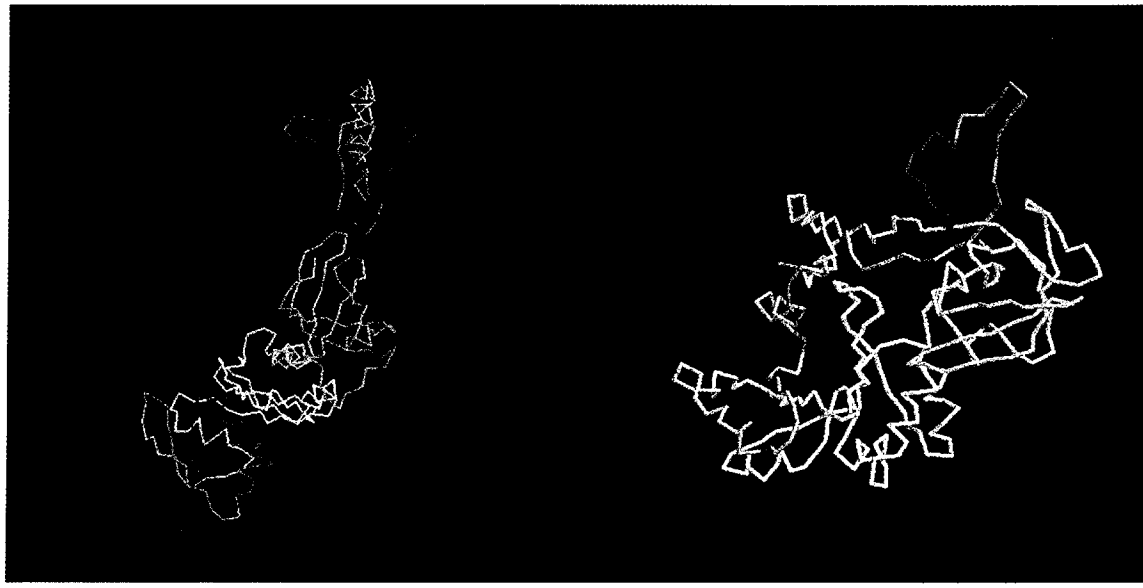


Figure 7.4. The quaternary model and the full experimental structure of NusA
The full experimental structure (left) compared with the quaternary model (right).

7.4 Discussion and conclusion

Because domains are considered as independent units of protein 3D structures (e.g. Richardson 1981), most current fold recognition programs are designed to recognise and model protein structure at the domain level. For simplicity, alignment algorithms try to detect similarity relationships for single domains, rather than align whole sequences and structures of multi-domain proteins. Most of these programs, like TUNE1D, use template libraries of protein domains.

Submitting the full sequence of a multi-domain protein to TUNE1D often leads to the detection of no appropriate templates. This occurs even when the global-local (Fisher 1996) or local (Smith and Waterman 1981) dynamic programming algorithms, which are less affected by the irrelevant parts of sequences. Normally, the best performance can be obtained by submitting the exact sequence segments corresponding to each domain.

However, prediction of domain boundaries from the sequences of multi-domain proteins is a challenging task. A manual and iterative approach was used here, considering the results of fold recognition for each segment. Different segmentations of the sequence were tried to assure most segments have good templates in our library. Obviously, a wrongly assigned boundary of one domain will lead to mistakes on boundary prediction for the other domains, and thus make fold recognition of the other domains more difficult as well. However, if we find a highly confident template for one segment, the boundary

prediction of its neighbours should benefit from this discovery. Since good templates for three of the four domains in NusA protein could be found, the domain boundary prediction here is relatively easy. If for a multi-domain protein, few of its domains have homologues in the template library, the prediction can be much harder. Nevertheless, as direct submision of the full sequence of NusA to TUNE1D leads to none of these templates being recognised, a domain boundary prediction procedure is clearly necessary.

Obviously, such a manual approach for domain boundary prediction is not practical in large-scale experiments. I am working with others in the Division of mathematical Biology on an algorithm to integrate results from iterative fold recognition of each segment and other analyses. We hope this program can be tested in the CASP and CAFASP experiments.

For NusA, TUNE1D detected good templates for the S1 domain and the two KH domains. The sequence identities of the alignments between the templates and the native domains are 22%, 16% and 12 %. It is worth noting that KH motif proteins can have one of two distinct global topologies even though they share significant sequence similarity (Grishin 2001). Here TUNE1D correctly found the type II KH domain template for the two highly similar domains. The structure of the N terminal domain is most similar to the B chain of the rad50cd ABC ATPase (PDB code 1f2t). This domain is classified as a fragment from multi-chain domains in CATH (CATH classification 9.1.160.1.1). Since it is regarded as preliminary data in CATH, it was not included in the TUNE1D template library. By definition, TUNE1D failed to detect it.

Even with the good models for three domains, the manually constructed quaternary model is still of poor quality. Relative orientations of domains were adjusted to make the model look similar to known multi-domain protein with similar function, without calculation of domain-domain or protein-RNA interaction. It is obvious that such a manual procedure is not optimal or robust. In NusA, the S1 and two KH modules which are very likely to be involved in RNA binding, are held together in a rigid arrangement. Conserved residues on the interfaces between the S1 and two KH domains strongly suggest that these three domains function together in RNA binding. On the other hand, the linker between the N terminal domain and S1 domain is more flexible. Relative orientation between them could be less conserved. Comparison of structures of the NusA proteins from *Mycobacterium tuberculosis* and *Thermotoga maritima* (Worbs et al. 2001) confirmed these assumptions. However, an automatic method for detecting contacts between different domains should be considered in an attempt to improve the performance of fold recognition programs on multi-domain proteins.

Overall, TUNE1D confidently detected good templates for three out of four domains in the NusA protein. The function of RNA binding of these domains can be easily predicted from the TUNE1D results. The domain boundary prediction is very successful as well. However, the completeness of the template library should be improved. Further, the process of domain boundary prediction, and the combination of different domains should be formulated and automated.

Chapter 8.

Summary

As stated before, when there are good templates in our protein structure libraries, good protein fold recognition methods become very useful to reveal structural information from protein sequences. Many studies have shown that by including structural information from templates, we can improve the accuracy of both alignment and detection of evolutionary relationship between protein sequences (e.g. Kelley et al. 2000, Rost 1995, Shi et al. 2001). Protein function annotation, detection of amino acid correlated mutations, protein secondary structure prediction, domain boundary prediction and many other applications can benefit from better fold recognition methods. Methods to find good templates and make accurate alignments can also be employed in comparative modelling programs, which can generate models of protein structure with adequate quality.

The vast amount of related publications and the increasing level of participation in the CASP and CAFASP experiments indicate the active research in the field of protein fold recognition. Overall participation has changed over the CASPs from 34 groups in CASP1, then 70, then 98, to 163 (including CAFASP2) in CASP4 (Moult et al. 2001). Most of them submitted models of fold recognition targets.

However, the progress of fold recognition methods in CASP experiments is rather disappointing. Alignment quality, the dominant factor affecting the accuracy of comparative modelling and fold recognition, has improved very little since CASP2. Although there is some evidence of advance, it could be due to the increasing data sets. Also, the accurate modelling of hard targets is still beyond the capability of current threading methods (Sippl et al. 1999, Marchler-Bauer and Bryant 1999, Venclovas et al. 2001).

Considering its simplicity, TUNE1D, the program we tested in CASP4, performed well as a template detector. The models often shared global similarity with the experimentally defined structures. However, the models are often of very limited quality. Human intervention, such as manual modification of alignments, is clearly necessary for better predictions.

The SMN model, described in the second chapter, is a simplified version of the TUNE1D model. Assuming that the PSSM using no structural information should converge to substitution matrices, I deliberately excluded structural environment description for the TUNE1D model, and found a new method for constructing substitution matrices. That my matrices are very similar to the proven BLOSUM and PAM matrices (Dayhoff et al. 1978, Henikoff and Henikoff 1992) is the first validation of my approach.

To integrate structural environment description into SMN models, I need a scheme to

characterise the structural environment of a residue. To improve speed and prevent over-fitting, the size of this scheme should be reduced. For this description, I developed a series of encoding schemes to describe amino acid types with a few real numbers. The AESNN schemes are directly obtained from the hidden layers of the SMN models. It is based on the simple observation that the hidden layer and the output layer of a SMN model is a (sub) neural network model as well. If a scheme of amino acid types performs well with this (sub) neural network, it could perform well on other neural network models, too. In this approach, the encoding schemes are automatically optimised by the training algorithm of neural network. We can easily build many schemes of different sizes. Examination of the schemes showed that their performance is good compared to other schemes and their compositions are clearly related to the physiochemical and evolutionary properties of amino acids. A colouring scheme of amino acid types was constructed using one of these schemes.

To improve the performance of threading programs, and to investigate different descriptions of residue structural environment, I introduced the TUNE model to measure protein sequence-structure compatibility. This model takes the residue structural environment description as input and predicts probabilities of observing amino acid types in such environments. Using this model, I generate a scoring function to measure the fitness of a residue in a protein model. By using an integrated structural environment description, my model outperformed traditional pseudo-energy functions. This framework, with the predefined input, target, scoring function and automatic optimisation algorithm, is very convenient for the testing of different structural environment

descriptions. The TUNE1D model then adapted these descriptions.

TUNE1D is the first fold recognition program in the TUNE package. With PSSMs generated from artificial neural network models, it makes 3D-1D matching using a dynamic programming algorithm. 3D-1D matching with PSSMs from structure templates is an established method for fold recognition (e.g. Johnson et al. 1993, Rice and Eisenberg 1997, Kelley et al. 2000, Shi et al. 2001). TUNE1D tested the new idea of using an artificial neural network to integrate structure environment descriptions to generate PSSMs. Although capable of using more complex structural environment description, TUNE1D is still a simple and fast method to generate PSSMs from a single protein template. TUNE1D is not a threading program. However, in the future threading program, matrices from TUNE1D will be employed for searching with the double dynamic programming algorithm (Taylor 1997a, Taylor and Orengo 1989, Jones et al. 1992a).

In chapter 7, we tested TUNE1D and other programs in a "mini CASP" experiment. We predicted the structure of NusA (N utilization Substance proteins A) protein from *Mycobacterium tuberculosis*. Again, as a template detector, TUNE1D performed well. However, human intervention played a vital role in domain boundary assignment, without which the fold recognition could not be successfully applied. The manual construction of tertiary structure was unsuccessful. Automatic fold recognition of multi-domain proteins remains a challenge.

This thesis has described the development of the TUNE. Hopefully, in the future, multiple alignment and threading algorithms will be implemented. Interfaces to other programs, such as protein secondary structure prediction, contact prediction, loop construction and domain boundary prediction, will be included. A web server of TUNE1D was constructed and will be improved. TUNE will be tested in future CASPs and CAFASPs. I hope it could become a useful software package for others.

Reference:

Abagyan, R. A. & Batalov, S. (1997). Do aligned sequences share the same fold? *Journal of Molecular Biology* **273**(1), 355-68.

Abola, E. E., Sussman, J. L., Prilusky, J. & Manning, N. O. (1997). Protein Data Bank archives of three-dimensional macromolecular structures. *Methods Enzymol* **277**, 556-71.

Alexandrov, N. N., Nussinov, R. & Zimmer, R. M. (1996). Fast protein fold recognition via sequence to structure alignment and contact capacity potentials. *Pac Symp Biocomput*, 53-72.

Altschul, S. F. (1991). Amino-acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology* **219**(3), 555-565.

Altschul, S. F. (1993). A protein alignment scoring system sensitive at all evolutionary distances. *Journal of Molecular Evolution* **36**(3), 290-300.

Altschul, S. F. & Erickson, B. W. (1986). Optimal sequence alignment using affine gap costs. *Bull Math Biol* **48**(5-6), 603-16.

Altschul, S. F. & Gish, W. (1996). Local alignment statistics. *Methods Enzymol* **266**, 460-80.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology* **215**(3), 403-410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**(17), 3389-402.

Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science* **181**(96), 223-30.

Aszodi, A. & Taylor, W. R. (1996). Homology modelling by distance geometry. *Fold Des* **1**(5), 325-34.

Avbelj, F. (1992). Use of a potential of mean force to analyze free-energy contributions in protein folding. *Biochemistry* **31**(27), 6290-6297.

Bairoch, A. & Apweiler, R. (1996). The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Res* **24**(1), 21-5.

Baker, D. & Šali, A. (2001). Protein structure prediction and structural genomics. *Science* **294**(5540), 93-6.

Baldi, P. & Brunak, S. (1998). *Bioinformatics - the machine learning approach*, The MIT Press, Cambridge, MA,.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. & Nielsen, H. (2000a). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* **16**(5), 412-424.

Baldi, P., Brunak, S., Frasconi, P., Soda, G. & Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics* **15**(11), 937-46.

Baldi, P., Pollastri, G., Andersen, C. A. & Brunak, S. (2000b). Matching protein beta-sheet partners by feedforward and recurrent neural networks. *Proc Int Conf Intell*

Syst Mol Biol **8**, 25-36.

Bates, P. A., Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. (2001). Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins* **45**(Suppl 5), 39-46.

Benner, S. A., Cohen, M. A. & Gonnet, G. H. (1994). Amino-acid substitution during functionally constrained divergent evolution of protein sequences. *Protein Engineering* **7**(11), 1323-1332.

Bienkowska, J. R., Rogers, R. G. & Smith, T. F. (1999). Performance of threading scoring function designed using new optimization method. *Journal of Computational Biology* **6**, 299-311.

Bishop, C. M. (1995). *Neural networks for pattern recognition*, Oxford University Press.

Blake, J. D. & Cohen, F. E. (2001). Pairwise sequence alignment below the twilight zone. *Journal of Molecular Biology* **307**(2), 721-35.

Bonneau, R., Tsai, J., Ruczinski, I., Chivian, D., Rohl, C., Strauss, C. E. & Baker, D. (2001). Rosetta in CASP4: Progress in ab initio protein structure prediction. *Proteins* **45**(Suppl 5), 119-26.

Bork, P. (1991). Shuffled domains in extracellular proteins. *FEBS Lett* **286**(1-2), 47-54.

Bowie, J. U., Luthy, R. & Eisenberg, D. (1991). A method to identify protein sequences that fold into a known 3-dimensional structure. *Science* **253**(5016), 164-170.

Brenner, S. E., Chothia, C. & Hubbard, T. J. P. (1998). Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences of the United States of America* **95**(11), 6073-6078.

Bryant, S. H. (1996). Evaluation of threading specificity and accuracy. *Proteins* **26**(2), 172-85.

Burns, C. M., Richardson, L. V. & Richardson, J. P. (1998). Combinatorial effects of NusA and NusG on transcription elongation and Rho-dependent termination in Escherichia coli. *Journal of Molecular Biology* **278**(2), 307-16.

Bystroff, C., Thorsson, V. & Baker, D. (2000). HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. *Journal of Molecular Biology* **301**(1), 173-90.

Chothia, C. & Lesk, A. (1986). The Relation between the divergence of sequence and structure in proteins. *EMBO Journal* **5**(4), 823-826.

Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res* **16**(22), 10881-90.

Cootes, A. P., Curmi, P. M. G., Cunningham, R., Donnelly, C., Torda, A. E. (1998) The dependence of amino acid pair correlations on structural environment. *Proteins* **32**(2), 175-189

Cristobal, S., Zemla, A., Fischer, D., Rychlewski, L. & Elofsson, A. (2001). A study of quality measures for protein threading models. *BMC Bioinformatics* **2**(1), 5.

Dayhoff, M. O. (1978). Atlas of protein sequence, Vol. 5.

Dietman, S. & Holm, L. (2001). Identification of homology in protein structure classification. *Nature Structure Biology* **8**(11), 953-957.

Durbin, R., Eddy, S., Krogh, A. & Mitchison, G. (1998). *Biological sequence analysis.* 1st edit, 1. 1 vols, Cambridge University Press.

Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics* **14**(9), 755-63.

Elofsson, A. (2002). A study on protein sequence alignment quality. *Proteins* **46**(3), 330-9.

Elofsson, A., Fischer, D., Rice, D. W., LeGrand, S. M. & Eisenberg, D. (1996). A study of combined structure/sequence profiles. *Folding & Design* **1**(6), 451-461.

Feng, D. F. & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution* **25**(4), 351-60.

Fischer, D., Barret, C., Bryson, K., Elofsson, A., Godzik, A., Jones, D., Karplus, K. J., Kelley, L. A., MacCallum, R. M., Pawowski, K., Rost, B., Rychlewski, L. & Sternberg, M. (1999). CAFASP-1: critical assessment of fully automated structure prediction methods. *Proteins* **Suppl**(3), 209-17.

Fischer, D. & Eisenberg, D. (1996). Protein fold recognition using sequence-derived predictions. *Protein Science* **5**(5), 947-55.

Fischer, D., Elofsson, A. & Rychlewski, L. (2000). The 2000 Olympic games of protein structure prediction; fully automated programs are being evaluated vis-a-vis human teams in the protein structure prediction experiment CAFASP2. *Protein Eng* **13**(10), 667-70.

Fischer, D., Elofsson, A., Rychlewski, L., Pazos, F., Valencia, A., Rost, B., Ortiz, A. R. & Dunbrack, R. L., Jr. (2001). CAFASP2: The second critical assessment of fully automated structure prediction methods. *Proteins* **45**(Suppl 5), 171-83.

Fischer, D., Rice, D., Bowie, J. U. & Eisenberg, D. (1996). Assigning amino acid sequences to 3-dimensional protein folds. *FASEB J* **10**(1), 126-36.

Frishman, D. & Argos, P. (1996). Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence. *Protein Engineering* **9**(2), 133-42.

Frishman, D. & Argos, P. (1997). Seventy-five percent accuracy in protein secondary structure prediction. *Proteins-Structure Function and Genetics* **27**(3), 329-335.

Gatchell, D. W., Dennis, S. & Vajda, S. (2000). Discrimination of near-native protein structures from misfolded models by empirical free energy functions. *PROTEINS: Structure Function, and Genetics* **41**(4), 518-534.

George, R. A. & Heringa, J. (2000). The REPRO server: finding protein internal sequence repeats through the Web. *Trends Biochem Sci* **25**(10), 515-7.

Gibson, T. J., Thompson, J. D. & Heringa, J. (1993). The KH domain occurs in a diverse set of RNA-binding proteins that include the antiterminator NusA and is probably involved in binding to nucleic acid. *FEBS Letters* **324**(3), 361-6.

Gonnet, G. H., Cohen, M. A. & Benner, S. A. (1992). Exhaustive matching of the entire protein sequence database. *Science* **256**(5062), 1443-5.

Gopal, B., Haire, L. F., Gamblin, S. J., Dodson, E. J., Lane, A. N., Papavinasasundaram, K. G., Colston, M. J. & Dodson, G. (2001). Crystal structure of the transcription elongation/anti-termination factor NusA from Mycobacterium tuberculosis at 1.7 A resolution. *Journal of Molecular Biology* **314**(5), 1087-95.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology* **162**(3), 705-8.

Gotoh, O. (1995). A weighting system and algorithm for aligning many phylogenetically related sequences. *Comput Appl Biosci* **11**(5), 543-51.

Greenblatt, J. & Li, J. (1981). Interaction of the sigma factor and the NusA gene protein of E. coli with RNA polymerase in the initiation-termination cycle of

transcription. *Cell* **24**(2), 421-8.
Grishin, N. V. (2001). KH domain: one motif, two folds. *Nucleic Acids Res* **29**(3), 638-43.
Head-Gordon, T. & Brooks, C. L. (1991). Virtual rigid body dynamics. *Biopolymers* **31**(1), 77-100.
Henikoff, S. & Henikoff, J. G. (1992). Amino-acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America* **89**(22), 10915-10919.
Henikoff, S. & Henikoff, J. G. (1993). Performance evaluation of amino-acid substitution matrices. *Proteins-Structure Function and Genetics* **17**(1), 49-61.
Henikoff, S. & Henikoff, J. G. (1994). Position-based sequence weights. *Journal of Molecular Biology* **243**(4), 574-8.
Henikoff, S. & Henikoff, J. G. (1997). Embedding strategies for effective use of information from multiple sequence alignments. *Protein Science* **6**(3), 698-705.
Heringa, J. (1999). Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment. *Comput Chem* **23**(3-4), 341-64.
Higgins, D. G. & Sharp, P. M. (1988). CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene* **73**(1), 237-44.
Holm, L. & Sander, C. (1991). Database algorithm for generating protein backbone and side- chain coordinates from a c-alpha trace application to model- building and detection of coordinate errors. *Journal of Molecular Biology* **218**(1), 183-194.
Holm, L. & Sander, C. (1992). Evaluation of protein models by atomic solvation preference. *Journal of Molecular Biology* **225**(1), 93-105.
Holm, L. & Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *J Mol Biol* **233**(1), 123-38.
Holm, L. & Sander, C. (1994). The FSSP database of structurally aligned protein fold families. *Nucleic Acids Res* **22**(17), 3600-9.
Holm, L. & Sander, C. (1998). Dictionary of recurrent domains in protein structures. *Proteins* **33**(1), 88-96.
Honig, B. (1999). Protein folding: from the levinthal paradox to structure prediction. *Journal of Molecular Biology* **293**(2), 283-93.
Huber, T., Russell, A. J., Ayers, D. & Torda, A. E. (1999). Sausage: protein threading with flexible force fields. *Bioinformatics* **15**(12), 1064-5.
Jagla, B. & Schuchhardt, J. (2000). Adaptive encoding neural networks for the recognition of human signal peptide cleavage sites. *Bioinformatics* **16**(3), 245-250.
Johnson, M. S. & Overington, J. P. (1993). A structural basis for sequence comparisons - an evaluation of scoring methodologies. *Journal of Molecular Biology* **233**(4), 716-738.
Johnson, M. S., Overington, J. P. & Blundell, T. L. (1993). Alignment and searching for common protein folds using a data bank of structural templates. *Journal of Molecular Biology* **231**(3), 735-52.
Jones, D. T. (1999a). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *Journal of Molecular Biology* **287**(4), 797-815.
Jones, D. T. (1999b). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology* **292**(2), 195-202.

Jones, D. T., Taylor, W. R. & Thornton, J. M. (1992a). A new approach to protein fold recognition. *Nature* **358**(6381), 86-89.

Jones, D. T., Taylor, W. R. & Thornton, J. M. (1992b). The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences* **8**(3), 275-282.

Jones, D. T. & Thornton, J. M. (1996). Potential energy functions for threading. *Current Opinion in Structural Biology* **6**(2), 210-216.

Kabsch, W. & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**(12), 2577-2637.

Karplus, K., Karchin, R., Barrett, C., Tu, S., Cline, M., Diekhans, M., Grate, L., Casper, J. & Hughey, R. (2001). What is the value added by human intervention in protein structure prediction? *Proteins* **45**(Suppl 5), 86-91.

Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. E. (2000). Enhanced genome annotation using structural profiles in the program 3D-PSSM. *Journal of Molecular Biology* **299**(2), 499-520.

Lackner, P., Koppensteiner, W. A., Domingues, F. S. & Sippl, M. J. (1999). Automated large scale evaluation of protein structure predictions. *Proteins* **37**(S3), 7-14.

Lackner, P., Koppensteiner, W. A., Sippl, M. J. & Domingues, F. S. (2000). ProSup: a refined tool for protein structure alignment. *Protein Eng* **13**(11), 745-52.

Lazaridis, T. & Karplus, M. (2000). Effective energy functions for protein structure prediction. *Current Opinion in Structural Biology* **10**(2), 139-145.

Lin, K., May, A. C. W. & Taylor, W. R. (2001). Amino acid substitution matrices from an artificial neural network model. *Journal of Computational Biology* **8**(5), 471-481.

Lin, K., May, A. C. W. & Taylor, W. R. (2002a). Amino acid encoding schemes from protein structural alignments: multi-dimensional vectors to describe residue types. *Journal of Theoretical Biology (in press)*

Lin, K., May, A. C. W. & Taylor, W. R. (2002b). Threading using neural networks (TUNE): the measure of protein sequence-structure compatibility. *Bioinformatics (in press)*

Lindahl, E. & Elofsson, A. (2000). Identification of related proteins on family, superfamily and fold level. *Journal of Molecular Biology* **295**(3), 613-25.

Lipman, D. J., Altschul, S. F. & Kececioglu, J. D. (1989). A tool for multiple sequence alignment. *Proc Natl Acad Sci U S A* **86**(12), 4412-5.

Lipman, D. J. & Pearson, W. R. (1985). Rapid and sensitive protein similarity searches. *Science* **227**(4693), 1435-41.

Lu, H. & Skolnick, J. (2001). A distance-dependent atomic knowledge-based potential for improved protein structure selection. *PROTEINS: Structure, Function, and Genetics* **44**, 223-232.

Maass, W. (1995). Vapnik-Chervonenkis dimension of neural nets. In *The Handbook of Brain Theory and Neural Networks* (Arbib, M. A., ed.), pp. 1000-1003. Bradford Books/MIT Press.

Mah, T. F., Kuznedelov, K., Mushegian, A., Severinov, K. & Greenblatt, J. (2000). The alpha subunit of E. coli RNA polymerase activates RNA binding by NusA. *Genes Dev* **14**(20), 2664-75.

Marchler-Bauer, A. & Bryant, S. H. (1999). A measure of progress in fold recognition? *Proteins* **Suppl**(3), 218-25.

May, A. C. (1996). Pairwise iterative superposition of distantly related proteins and assessment of the significance of 3-D structural similarity. *Protein Engineering* **9**(12), 1093-1101.

May, A. C. (1999). Towards more meaningful hierarchical classification of amino acid scoring matrices. *Protein Eng* **12**(9), 707-12.

May, A. C. (2001). Optimal classification of protein sequences and selection of representative sets from multiple alignments: application to homologous families and lessons for structural genomics. *Protein Eng* **14**(4), 209-17.

May, A. C. & Blundell, T. L. (1994). Automated comparative modelling of protein structures. *Curr Opin Biotechnol* **5**(4), 355-60.

May, A. C. & Johnson, M. S. (1994). Protein structure comparisons using a combination of a genetic algorithm, dynamic programming and least-squares minimization. *Protein Eng* **7**(4), 475-85.

McClure, M. A., Vasi, T. K. & Fitch, W. M. (1994). Comparative-Analysis of multiple protein-sequence alignment Methods. *Molecular Biology and Evolution* **11**(4), 571-592.

Miller, W. & Myers, E. W. (1988). Sequence comparison with concave weighting functions. *Bull Math Biol* **50**(2), 97-120.

Morgenstern, B., Dress, A. & Werner, T. (1996). Multiple DNA and protein sequence alignment based on segment-to-segment comparison. *Proc Natl Acad Sci U S A* **93**(22), 12098-103.

Mosimann, S., Meleshko, R. & James, M. N. G. (1995). A critical-assessment of comparative molecular modeling of tertiary structures of proteins. *PROTEINS: Structure, Function, and Genetics* **23**(3), 301-317.

Moult, J. (1997). Comparison of database potentials and molecular mechanics force fields. *Curr Opin Struct Biol* **7**(2), 194-9.

Moult, J., Hubbard, T., Fidelis, K. & Pedersen, J. T. (1999). Critical assessment of methods of protein structure prediction (CASP): Round III. *Proteins-Structure Function and Genetics*, 2-6.

Moult, J., Judson, R., Fidelis, K. & Pedersen, J. T. (1995). A large-scale experiment to assess protein structure prediction methods. *PROTEINS: Structure, Function, and Genetics* **23**, ii-iv.

Moult, J. & Unger, R. (1991). An analysis of protein folding pathways. *Biochemistry* **30**(16), 3816-3824.

Muller, K., Finke, M., Schulten, K., Murata, N. & Amari, S. (1996). A numerical study on learning curves in stochastic multi-layer feed-forward networks. *Neural Computation* **8**, 1085-1106.

Murzin, A. G. & Bateman, A. (2001). CASP2 knowledge-based approach to distant homology recognition and fold prediction in CASP4. *Proteins* **45**(Suppl 5), 76-85.

Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). SCOP: a structural classification of proteins database for the invertigation of sequences and structures. *Journal of Molecular Biology* **241**(4), 536-540.

Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular*

Biology **48**, 443-53.
Notredame, C., Higgins, D. G. & Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol* **302**(1), 205-17.
Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. & Thornton, J. M. (1997). CATH - a hierarchic classification of protein domain structures. *Structure* **5**, 1093-1108.
Orengo, C. A., Sillitoe, I., Reeves, G. & Pearl, F. M. (2001). Review: what can structural classifications reveal about protein evolution? *Journal of Structural Biology* **134**(2-3), 145-65.
Overington, J., Donnelly, D., Johnson, M. S., Šali, A. & Blundell, T. L. (1992). Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds. *Protein Science* **1**(2), 216-226.
Park, B. & Levitt, M. (1996). Energy functions that discriminate X-ray and near-native folds from well-constructed decoys. *Journal of Molecular Biology* **258**(2), 367-392.
Park, B. H. & Levitt, M. (1995). The complexity and accuracy of discrete state models of protein structure. *Journal of Molecular Biology* **249**(2), 493-507.
Pearson, W. R. & Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A* **85**(8), 2444-8.
Peitsch, M. C. (1996). ProMod and Swiss-Model: Internet-based tools for automated comparative protein modelling. *Biochem Soc Trans* **24**(1), 274-9.
Qian, N. & Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology* **202**(4), 865-84.
Ramachandran, G. N., Kolaskar, A. S., Ramakrishnan, C. & Sasisekharan, V. (1974). The mean geometry of the peptide unit from crystal structure data. *Biochim Biophys Acta* **359**(2), 298-302.
Rice, D. W. & Eisenberg, D. (1997). A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. *Journal of Molecular Biology* **267**(4), 1026-1038.
Richardson, J. S. (1981). The anatomy and taxonomy of protein structure. *Adv Protein Chem* **34**, 167-339.
Risler, J., Delorme, M., DELACROIX, H. & HENAUT, A. (1988). Amino-acid subsitution in structurally related proteins - a pattern-recognition approach - determination of a new and efficient scoring matrix. *Journal of Molecular Biology* **204**(4), 1019-1029.
Rost, B. (1995). TOPITS: threading one-dimensional predictions into three-dimensional structures. *Proc Int Conf Intell Syst Mol Biol* **3**, 314-21.
Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Eng* **12**(2), 85-94.
Rost, B. & Sander, C. (1993). Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc Natl Acad Sci U S A* **90**(16), 7558-62.
Rost, B. & Sander, C. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins* **19**(1), 55-72.
Rost, B., Schneider, R. & Sander, C. (1997). Protein fold recognition by prediction-based threading. *Journal of Molecular Biology* **270**(3), 471-480.

Rufino, S. D., Donate, L. E., Canard, L. H. & Blundell, T. L. (1997). Predicting the conformational class of short and medium size loops connecting regular secondary structures: application to comparative modelling. *Journal of Molecular Biology* **267**(2), 352-67.

Russell, R., Saqi, M., Sayle, R., Bates, P. & Sternberg, M. (1997). Recognition of analogous and homologous protein folds: Analysis of sequence and structure conservation. *Journal of Molecular Biology* **269**(3), 423-439.

Russell, R. B., Copley, R. R. & Barton, G. J. (1996). Protein fold recognition by mapping predicted secondary structures. *Journal of Molecular Biology* **259**(3), 349-65.

Rychlewski, L., Jaroszewski, L., Li, W. & Godzik, A. (2000). Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Sci* **9**(2), 232-41.

Salamov, A. A. & Solovyev, V. V. (1997). Protein secondary structure prediction using local alignments. *Journal of Molecular Biology* **268**(1), 31-6.

Samudrala, R. & Moult, J. (1998). An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *Journal of Molecular Biology* **275**(5), 895-916.

Sander, C. & Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins* **9**(1), 56-68.

Schneider, T. D. (1997). Information content of individual cenetic sequences. *Journal of Theoretical Biology* **189**(4), 427-441.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal* **27**, 379-423, 623-656.

Shi, J., Blundell, T. L. & Mizuguchi, K. (2001). FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal of Molecular Biology* **310**(1), 243-57.

Simons, K. T., Bonneau, R., Ruczinski, I. I. & Baker, D. (1999). Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins* **37**(S3), 171-176.

Simons, K. T., Kooperberg, C., Huang, E. & Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of Molecular Biology* **268**(1), 209-25.

Sippl, M. J. (1995). Knowledge-based potentials for proteins. *Current Opinion in Structural Biology* **5**(2), 229-235.

Sippl, M. J., Lackner, P., Domingues, F. S. & Koppensteiner, W. A. (1999). An attempt to analyse progress in fold recognition from CASP1 to CASP3. *Proteins* **37**(S3), 226-230.

Skolnick, J., Jaroszewski, L., Kolinski, A. & Godzik, A. (1997a). Derivation and testing of pair potentials for protein folding. When is the quasichemical approximation correct? *Protein Science* **6**, 676-688.

Skolnick, J. & Kolinski, A. (1991). Dynamic Monte Carlo simulations of a new lattice model of globular protein folding, structure and dynamics. *Journal of Molecular Biology* **221**(2), 499-531.

Skolnick, J., Milik, M. & Kolinski, A. (1997b). Prediction of relative binding motifs of biologically active peptides and peptide mimetics. The Scripps Research Institute,

United States Patent.
Smith, A. V. & Hall, C. K. (2001). Protein refolding versus aggregation: computer simulations on an intermediate-resolution protein model. *Journal of Molecular Biology* **312**(1), 187-202.
Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology* **147**, 195-197.
Solis, A. D. & Rackovsky, S. (2000). Optimized representations and maximal information in proteins. *PROTEINS: Structure Function, and Genetics* **38**(2), 149-164.
Sussman, J. L., Lin, D., Jiang, J., Manning, N. O., Prilusky, J., Ritter, O. & Abola, E. E. (1998). Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. *Acta Crystallogr D Biol Crystallogr* **54**(1 (Pt 6)), 1078-84.
Sutcliffe, M. J., Haneef, I., Carney, D., Blundell, T. L. (1987). Knowledge based modelling of homologous proteins, Part I: Three-dimensional frameworks derived from the simultaneous superposition of multiple structures. *Protein Engineering* **1**(5), 377-84.
Taylor, W. R. (1986). The classification of amino-acid conservation. *Journal of Theoretical Biology* **119**(2), 205-218.
Taylor, W. R. (1987). Multiple sequence alignment by a pairwise algorithm. *Comput Appl Biosci* **3**(2), 81-7.
Taylor, W. R. (1988). A flexible method to align large numbers of biological sequences. *Journal of Molecular Evolution* **28**(1-2), 161-9.
Taylor, W. R. (1990). Hierarchical method to align large numbers of biological sequences. *Methods Enzymol* **183**, 456-74.
Taylor, W. R. (1997a). Multiple sequence threading: an analysis of alignment quality and stability. *Journal of Molecular Biology* **269**(5), 902-943.
Taylor, W. R. (1997b). Residual colours: a proposal for aminochromography. *Protein Engineering* **10**(7), 743-746.
Taylor, W. R. (1998). Dynamic sequence databank searching with templates and multiple alignment. *Journal of Molecular Biology* **280**(3), 375-406.
Taylor, W. R. (1999). Protein structure comparison using iterated double dynamic programming. *Protein Science* **8**(3), 654-665.
Taylor, W. R. & Brown, N. P. (1999). Iterated sequence databank search methods. *Comput Chem* **23**(3-4), 365-85.
Taylor, W. R. & Orengo, C. A. (1989). Protein structure alignment. *Journal of Molecular Biology* **208**(1), 1-22.
Thiele, R., Zimmer, R. & Lengauer, T. (1999). Protein threading by recursive dynamic programming. *Journal of Molecular Biology* **290**, 757-779.
Thornton, J. M., Orengo, C. A., Todd, A. E. & Pearl, F. M. (1999). Protein folds, functions and evolution. *Journal of Molecular Biology* **293**(2), 333-42.
Venclovas, Zemla, A., Fidelis, K. & Moult, J. (2001). Comparison of performance in successive CASP experiments. *Proteins* **45**(Suppl 5), 163-70.
Vendruscolo, M., Najmanovich, R. & Domany, E. (2000). Can a pairwise contact potential stabilize native protein folds against decoys obtained by threading? *PROTEINS: Structure, Function, and Genetics* **38**, 134-148.
Weisberg, R. A. & Gottesman, M. E. (1999). Processive antitermination. *Journal of*

Bacteriol **181**(2), 359-67.
Weiss, O. & Herzel, H. (1998). Correlations in protein sequences and property codes. *Journal of Theoretical Biology* **190**(4), 341-353.
Wetlaufer, D. B. (1973). Nucleation, rapid folding, and globular intrachain regions in proteins. *Proc Natl Acad Sci U S A* **70**(3), 697-701.
Williams, M. G., Shirai, H., Shi, J., Nagendra, H. G., Mueller, J., Mizuguchi, K., Miguel, R. N., Sc, S. C., Innis, C. A., Deane, C. M., Chen, L., Campillo, N., Burke, D. F., Blundell, T. L. & de Bakker, P. I. (2001). Sequence-structure homology recognition by iterative alignment refinement and comparative modeling. *Proteins* **45**(Suppl 5), 92-7.
Worbs, M., Bourenkov, G. P., Bartunik, H. D., Huber, R. & Wahl, M. C. (2001). An extended RNA binding surface through arrayed S1 and KH domains in transcription factor NusA. *Mol Cell* **7**(6), 1177-89.
Wu, C. H. & McLarty, J. M. (2000). *Neural network and genome informatics.* Methods in Computational Biology and Biochemistry (Konopka, A. K., Ed.). 1 vols, Elsevier Science.
Xu, Y. & Xu, D. (2000). Protein threading using PROSPECT: design and evaluation. *Proteins* **40**(3), 343-54.
Zemla, A., Venclovas, Moult, J. & Fidelis, K. (2001). Processing and evaluation of predictions in CASP4. *Proteins* **45**(Suppl 5), 13-21.
Zhang, K. Y. J. & Eisenberg, D. (1994). The 3-dimensional profile method using residue preference as a continuous function of residue environment. *Protein Science* **3**(4), 687-695.
Zhou, Y., Mah, T. F., Yu, Y. T., Mogridge, J., Olson, E. R., Greenblatt, J. & Friedman, D. I. (2001). Interactions of an Arg-rich region of transcription elongation protein NusA with NUT RNA: implications for the order of assembly of the lambda N antitermination complex in vivo. *Journal of Molecular Biology* **310**(1), 33-49.